

ПУГАЧ Тетяна Валеріївна

**Математичне та програмне забезпечення аналізу
серцевого ритму за відеоданими з мобільного
пристрою / Mathematical tools and software for
analysis the heart rate from video data from a
mobile device**

спеціальність: 121 - Інженерія програмного забезпечення
освітньо-професійна програма - Інженерія програмного забезпечення

Кваліфікаційна робота

Виконала студентка групи
ІПЗзмл-21
Т. В. Пугач

Науковий керівник:
к.е.н., доцент, Л. І. Гончар

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ОЦІНКА ФІЗІОЛОГІЧНИХ ПАРАМЕТРІВ ОРГАНІЗМУ ЛЮДИНИ.....	6
1.1. Аналіз сучасної ситуації та тенденцій розвитку захворювань серцево-судинної системи.....	6
1.2. Показники діяльності серцево-судинної системи, їх вплив на оцінку загального стану організму та методи їхнього визначення	12
1.3. Методи аналізу варіабельності серцевого ритму.....	23
1.3.1. Статистичні методи.....	26
1.3.2. Геометричні методи (варіаційна пульсометрія)	28
1.3.3. Автокореляційний аналіз	28
1.3.4. Кореляційна ритмографія – скаттерографія.....	29
1.3.5. Інші методи аналізу варіабельності серцевого ритму.....	30
Висновки до розділу 1	33
РОЗДІЛ 2 МАТЕМАТИЧНІ ТА ПРОГРАМНІ АСПЕКТИ РЕАЛІЗАЦІЇ АЛГОРИТМУ ОЦІНКИ ПОКАЗНИКІВ ДІЯЛЬНОСТІ СЕРЦЕВО-СУДИННОЇ СИСТЕМИ	35
2.1. Особливості використання методу фотоплетизмографії під час дослідження варіабельності серцевих скорочень з використанням камери смартфона	35
2.2. Попередній алгоритм роботи обчислювального конвеєру	38
2.3. Математична модель реалізації алгоритму вимірювання варіабельності серцевого ритму	41
2.4. Особливості застосування програмних рішень	44
Висновки до розділу 2	50
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ АНАЛІЗУ ВІДЕОДАНИХ	52
3.1. Особливості розробленого програмного забезпечення.....	52

	2
3.2. Підсистема обробки та зберігання даних	60
3.3. Статистична оцінка показників, одержаних роботи в результаті роботи програми.....	67
Висновки до розділу 3	72
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТКИ.....	82

ВСТУП

Захворювання серцево-судинної системи становлять, за статистикою, найвищу причину смертності в усьому світі. Через ці захворювання в країнах Європи щорічно помирає понад 4,3 мільйона людей, що складає 48% від світових показників. За даними Держстату, в Україні показник смертності від захворювань серцево-судинної системи значно вищий і становить 60%. Саме тому цій проблемі приділяється значна увага.

Наукові та клінічні дослідження в галузі кардіології постійно розвиваються, що дає поштовх для виникнення сучасних способів діагностики та лікування різних захворювань і дає змогу боротися з хворобами різного походження. Проте, традиційна медицина має реактивний характер, тобто, пацієнти звертаються до медичних працівників не з профілактичною метою, а з уже вираженими ознаками захворювання. Глобальним трендом в сучасній медицині є профілактичний, персоналізований та прогностичний підхід, що передбачає регулярне спостереження за станом здоров'я людини ще до виникнення перших проявів хвороби.

Для точної діагностики серцево-судинних захворювань застосовують аналіз імпульсів електрокардіограми, що допомагає спрогнозувати виникнення серцево-судинної патології. Вимірювання серцевої активності таким способом передбачає встановлення на пацієнта кардіорекодерів з наступною інтерактивною обробкою лікарем отриманих даних.

Іншим методом діагностики захворювань серцево-судинної системи є дослідження варіабельності серцевого ритму, показника, який враховує вплив дихальної та нервової системи на роботу серця. За останні роки все більшої популярності набувають різноманітні альтернативи проведенню електрокардіограми, зокрема, фітнес-трекери, різноманітні пульсометри, «розумні» годинники, тощо. Переважна більшість цих пристроїв працюють на принципах фотоплетизмографії та передбачають контакт зі шкірою людини. Широкому розповсюдженню цих пристроїв заважає висока ціна, крім того,

часто стається так, що якісь сигнали для діагностики захворювання у багатьох випадках є незадовільною. Саме тому виникає необхідність в пошуках альтернативних джерел самодіагностики.

Метою кваліфікаційної роботи є розробка архітектури та реалізація програмної системи аналізу відеоданих для визначення показників діяльності серцево-судинної системи.

Відповідно до мети роботи необхідно вирішити наступні **завдання**:

1. Здійснити огляд методів визначення показників роботи серцево-судинної системи, таких як частота серцевих скорочень та варіабельність серцевого ритму.
2. Спроекувати концептуальну модель програмного забезпечення для аналізу відеоданих.
3. Обрати математичний апарат для виділення та аналізу сигналу, одержаного шляхом перетворення відеоданих.
4. Здійснити огляд та побудову програмного стеку технологій для реалізації системи.
5. Побудувати модель зберігання даних для забезпечення функціонування програмної системи.
6. На основі вибраного стеку технологій розробки реалізувати програмне забезпечення для аналізу відеоданих, одержаних за допомогою камери смартфона, та визначення показників діяльності серцево-судинної системи.
7. Виконати порівняльний аналіз ефективності визначення показників діяльності серцево судинної системи за допомогою камери смартфона з даними, одержаним за допомогою альтернативних засобів вимірювання, зокрема смарт-годинників.

Об'єктом дослідження є алгоритмічне, математичне та програмне забезпечення аналізу відеоданих та визначення показників діяльності серцево-судинної системи. **Предметом** дослідження є програмна система для аналізу відеоданих, одержаних за допомогою камери смартфона.

Методи дослідження засновані на використанні алгоритму швидкого перетворення Фур'є при аналізі частотної складової сигналу, математичного та

програмного апарату перетворення колірних просторів, засобів бібліотек `numpy` та `opencv` мови програмування Python.

Новизна роботи полягає у подальшому розвитку сфери застосування алгоритму швидкого перетворення Фур'є до аналізу відеоданих, одержаних за допомогою камери смартфона, розробці методики обчислення показників роботи серцево-судинної системи шляхом аналізу сигналу, виділеного з відеоряду.

Практичне значення одержаних результатів. Реалізовано програмну систему аналізу відеоданих та розрахунку показників роботи серцево-судинної системи, таких як частота серцевих скорочень та варіабельність серцевого ритму. Також програмний продукт дозволяє вибудовувати кардіоритмограму для спостереження динаміки роботи серцево-судинної системи. Додаток може бути застосований для моніторингу стану здоров'я як альтернатива пристроям, оснащеним спеціалізованими датчиками та сенсорами.

Публікації та апробація результатів.

1. Ройко О.Ю., Пугач Т.В. Особливості реалізації програмного забезпечення для визначення показників роботи серця з використанням камери смартфона. Комп'ютерно-інтегровані технології: освіта, наука, виробництво, №48, 2022. С. 141-146

РОЗДІЛ 1

ОЦІНКА ФІЗІОЛОГІЧНИХ ПАРАМЕТРІВ ОРГАНІЗМУ ЛЮДИНИ

1.1. Аналіз сучасної ситуації та тенденцій розвитку захворювань серцево-судинної системи

Ретельний аналіз тенденцій впливу провідних факторів ризику та оцінка їхнього впливу на здоров'я людини є важливими для визначення того, де загальноприйнята система охорони здоров'я досягає прогресу, а в яких випадках поточні зусилля є недостатніми.

Серцево-судинні захворювання, серед яких переважно ішемічна хвороба серця та інсульт, є основними причинами смертності й одними з основних факторів інвалідності в усьому світі. Такі висновки були отримані в дослідженні Глобального тягара хвороб за 2019 рік [1]. Global Burden of Disease – це дослідження, під час якого відбувається дослідження показників проблем зі здоров'ям та рівнем смертності серед населення планети, починаючи з 1990 року, у понад 195 країнах, включно з Україною. Це дослідження охоплює понад 400 різних травм, факторів ризику та хвороб. Дане дослідження координується Інститутом метрики та оцінювання в системі охорони здоров'я при Вашингтонському університеті, а його результати публікуються раз на декілька років. Останній такий звіт був виданий у 2019 році.

Згідно з висновками Глобального тягара хвороб, рівень захворюваності серцево-судинної системи продовжує стрімко зростати протягом десятиліть майже у всіх країнах із середнім та низьким рівнем доходів населення, при чому, стандартизований за віком показник серцево-судинних захворювань почав рости в деяких країнах із високим рівнем доходів населення, де раніше цей показник знижувався [1].

Серцево-судинні захворювання, згідно з даним дослідженням, спричиняють більше 60% смертей серед усіх смертельних випадків. Виявлення випадків серцево-судинних захворювань майже подвоїлося з 271

мільйона в 1990 році до 523 мільйонів у 2019 році, а кількість смертей від серцево-судинних захворювань у досліджуваний період неухильно збільшувалася з 12,1 мільйона в 1990 році до 18,6 мільйона у 2019 році. За прогнозами, у 2030 році смертність від серцево-судинних захворювань становитиме близько 25 мільйонів осіб на рік.

Згідно з результатами дослідження Глобального тягара хвороб, в Україні серцево-судинні захворювання є головною причиною смертності населення. За цим показником наша країна лишається одним зі світових лідерів [2].

	Ischemic heart disease	Stroke	Self-harm	Cirrhosis	Road injuries	Low back pain	Cardiomyopathy	Neonatal disorders	Alcohol use disorders	Falls
Ukraine	1	2	3	4	5	6	7	8	9	10
Comparison group mean (High-middle SDI)	1	4	14	23	10	2	47	11	28	13
Belarus	1	2	5	11	7	4	16	12	3	6
Bosnia and Herzegovina	1	2	14	22	10	4	24	6	26	7
Bulgaria	1	2	16	7	8	4	45	10	31	9
Georgia	1	2	25	6	5	9	49	3	28	10
Kazakhstan	1	2	5	4	7	9	11	6	12	16
North Macedonia	2	1	24	25	8	5	18	4	33	10
Romania	1	2	19	4	8	3	15	9	26	7
Serbia	1	2	12	25	10	4	22	7	23	8

Рис. 1.1. Десять найпоширеніших причин смертності та інвалідності серед країн з приблизно однаковими соціально-демографічними показниками, згідно з результатами дослідження Global Burden of Disease [2]

За даними Держстату України [3; 4] за дев'ять місяців 2021 року в Україні померло 487,9 тис. громадян, найбільший показник смертності спричинений саме захворюваннями серцево-судинної системи (306127 громадян). На другому місці серед поширених причин – онкологія, а на третьому місці серед причин смертності став коронавірус. Також серед найбільш поширених причин смерті – нещасні випадки. У національному масштабі смертність від серцево-судинних захворювань за останні 29 років зросла майже на 8 %.

Основними серцево-судинними захворюваннями, що спричиняють високий показник смертності є ішемічна хвороба серця (213278 громадян), цереброваскулярні хвороби (59270 громадян) та алкогольна кардіоміопатія (2549 громадян) [4].

Ранжування смертельних випадків, залежно від типу серцево-судинних захворювань серед чоловіків і жінок усіх вікових категорій подано у табл.1.1.

Таблиця 1.1.

Ранжування смертельних випадків, залежно від типу серцево-судинних захворювань серед чоловіків і жінок

№	Чоловіки	Жінки
1.	Ішемічна хвороба серця	Ішемічна хвороба серця
2.	Цереброваскулярні захворювання	Цереброваскулярні захворювання
3.	Кардіоміопатія і міокардит	Кардіоміопатія і міокардит
4.	Захворювання периферичних судин	Миготлива аритмія
5.	Аневризма аорти	Захворювання периферичних судин
6.	Миготлива аритмія	Інші серцево-судинні захворювання
7.	Інші серцево-судинні захворювання	Гіпертонічна хвороба серця
8.	Гіпертонічна хвороба серця	Ревмокардит
9.	Ревмокардит	Аневризма аорти

10.	Ендокардит	Неревматичні розлади клапанів
-----	------------	-------------------------------

В дослідженні [5] наведено прогноз поширеності захворювань серцево-судинної системи в Україні до 2025 року (на 100 тис. нас.), графічне представлення якого наведено на рис.1.2.

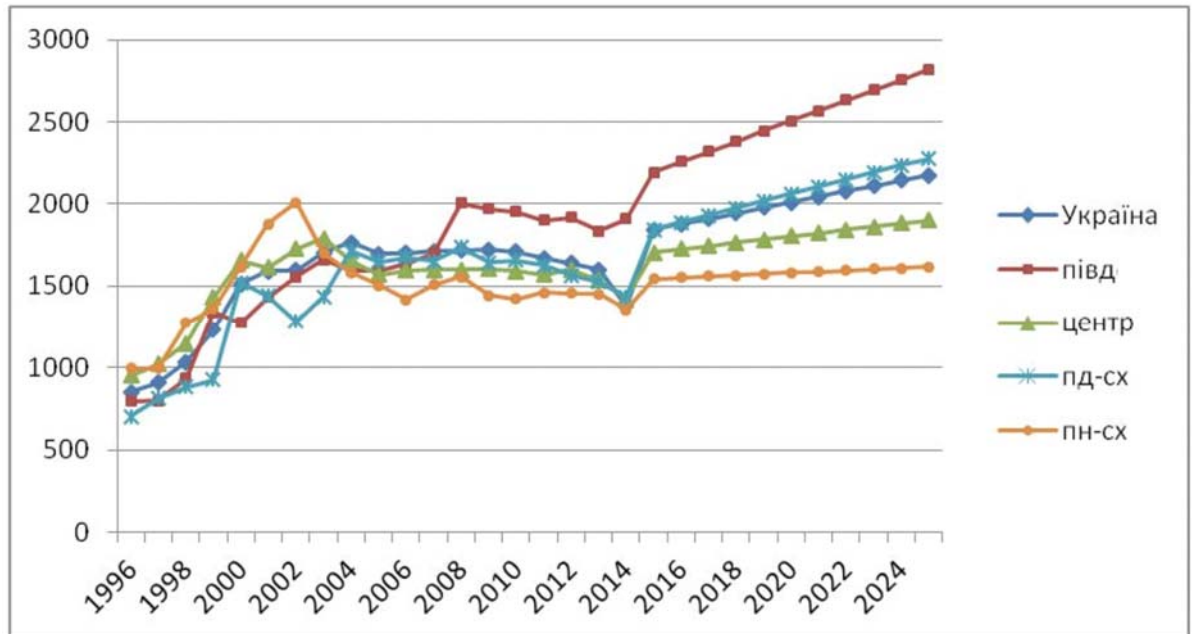


Рис.1.2. Прогноз динаміки серцево-судинної захворюваності в Україні та районах до 2025 року (на 100 тис. нас.) [5]

Рання діагностика серцево-судинних патологій сприяє своєчасному виявленню та лікуванню даних захворювань. Серед основних факторів ризику, що сприяють випадкам передчасної смерті від серцево-судинних захворювань виділяють:

1. Високий артеріальний тиск для людей віком від 45 років та понаднормовий індекс маси тіла.
2. Тютюновий дим як від пасивного так і активного куріння.
3. Надмірне споживання алкоголю.
4. Забрудненість повітря та навколишнього середовища.
5. Порушення функцій роботи нирок.
6. Незбалансоване харчування та високий рівень глюкози в крові.
7. Низька фізична активність та психологічний стрес.

8. Обтяжена спадковість, тобто наявність серцево-судинних захворювання у батьків, що виникли у працездатному віці [6].

Розвиток атеросклерозу, що призводить до розвитку ішемічної хвороби серця, може роками мати безсимптомний перебіг і проявитися на стадії ускладнень – інфаркту міокарда, мозкового інсульту, серцевої недостатності або призвести до раптової смерті. Важливо зауважити, що в Україні третина інфарктів та інсультів виникає серед осіб працездатного віку, що в десять разів перевищує відповідний показник для країн Західної Європи. Зниження передчасної смертності від серцево-судинних захворювань в Європі пов'язане з широким проведенням оздоровчих програм, активною участю населення в подоланні факторів ризику, підвищенням відповідальності за своє здоров'я [6].

Більшість факторів ризику залежать від стилю життя. Корекція факторів ризику важлива в усіх вікових категоріях, починаючи з дитинства.

Важливим фактором ранньої діагностики серцево-судинних захворювань є моніторинг рівня артеріального тиску. Гіпертонічна хвороба може мати безсимптомний перебіг, проте її наявність значно підвищує ризик розвитку порушень мозкового кровообігу, тобто інсульту та інфаркту міокарда. Рекомендується [6; 7] регулярно вимірювати артеріальний тиск. Якщо він перевищує нормальні показники, можна вважати, що це один з перших сигналів для усвідомлення ризиків виявлення серцево-судинних захворювань. Регулярний прийом препаратів для лікування артеріальної гіпертензії, підтримка оптимального рівня артеріального тиску є доведеним важливим заходом профілактики розвитку серцево-судинних захворювань.

Також, починаючи з сорока років, рекомендується проводити контроль рівня холестерину у крові. Для профілактики підвищення рівня холестерину важливо дотримуватись засад здорового харчування та рекомендованого медиками рівня фізичної активності впродовж всього життя. За наявності декількох різних факторів ризику, а також пацієнтам, які перенесли інфаркт

міокарда чи мозковий інсульт, для нормалізації рівня холестерину призначають фармакологічні препарати.

Як зазначалося вище, тютюнокуріння є одним із серйозних факторів ризику розвитку серцево-судинних захворювань та їх ускладнень. Ризик виникнення ішемічної хвороби серця у курців у 4 рази вищий ніж у тих, хто не курить, тому відмова від цієї шкідливої звички дуже важлива у будь-якому віці [8; 9].

Також, вагомим фактором, що призводить до розвитку ішемічної хвороби серця та інфаркту міокарда, є порушення обміну глюкози та цукровий діабет. З метою ранньої діагностики цукрового діабету рекомендується щорічне визначення рівня глюкози у крові. Основними профілактичними заходами розвитку цукрового діабету є здорове харчування, фізична активність та підтримка нормальної маси тіла.

Ожиріння, особливо абдомінальне ожиріння, коли збільшується обсяг талії, сприяє розвитку гіпертонічної хвороби, цукрового діабету, атеросклерозу та ішемічної хвороби серця. Рекомендовано [10] розрахувати, самостійно або разом з лікарем, свій індекс маси тіла. Для цього можна скористатися формулою (1.1)

$$\text{ІМТ} = \frac{\text{МТ}}{\text{ЗР}^2} \quad (1.1)$$

де

ІМТ – індекс маси тіла;

МТ – маса тіла в кілограмах;

ЗР – зріст людини в метрах.

В нормі величина індексу маси тіла становить 18-25 кг/м² [11]. За наявності понаднормового показника індексу маси тіла слід проаналізувати харчування та підвищити рівень фізичної активності.

Малорухомий спосіб життя – це ще одна проблема сучасного суспільства, наслідки якої мають серйозний вплив на організм людини. Недостатня фізична активність прискорює старіння організму, сприяє

розвитку ожиріння, серцево-судинних захворювань та цукрового діабету. Мінімальний рівень фізичної активності, який рекомендує Всесвітня організація охорони здоров'я – швидка хода по 30 хв. щоденно. В нормі, у дорослих, віком 18-64 роки повинно бути щонайменше 150 хвилин аеробної фізичної активності помірної інтенсивності протягом тижня або щонайменше 75 хвилин аеробної фізичної активності високої інтенсивності [12]. Корисними будуть регулярні прогулянки та ранкова гімнастика.

Хронічні психоемоційні перевантаження, стресові ситуації, високий рівень тривожності та депресія також мають вплив на виникнення та розвиток серцево-судинних захворювань. Важливо враховувати, що поєднання декількох факторів ризику значно прискорює розвиток серцево-судинних захворювань. Виявити фактори ризику можна при проведенні профілактичних оглядів. За результатами огляду лікар зможе зробити висновки щодо ризику розвитку серцево-судинних захворювань та визначити заходи по корекції факторів ризику.

У відповідності до сучасних рекомендацій, профілактику серцево-судинних захворювань потрібно починати з народження і проводити впродовж всього життя [9; 12].

1.2. Показники діяльності серцево-судинної системи, їх вплив на оцінку загального стану організму та методи їхнього визначення

Функціональний стан серцево-судинної системи організму є одним з основних показників функціональних можливостей людини. Оцінка стану серцево-судинної системи організму людини відіграє першочергове значення у зв'язку з немалою роллю даної системи у оптимальному функціонуванні організму в найрізноманітніших за своїм змістом видах діяльності. Комплексний підхід до оцінки функціонального стану серцево-судинної системи передбачає проведення реєстрації основних фізіологічних параметрів системи кровообігу у стані відносного спокою в поєднанні з аналізом їх реакції на дозовані тестувальні навантаження [13]. Отже, комплекс методів оцінки

функціонального стану серцево-судинної системи включає в себе наступні методи [7]:

1. Традиційні методи визначення інтегральних показників системи кровообігу, зокрема частоти серцевих скорочень, рівня артеріального тиску, систолічного об'єму крові, хвилинного об'єму кровообігу, серцевого індексу та інші показники.

2. Розрахункові методи визначення інтегральних параметрів серцево-судинної системи, зокрема, варіабельності серцевого ритму та інших.

3. Функціональні проби системи кровообігу, за допомогою яких оцінюється тип реакції апарату кровообігу на дозоване фізичне навантаження.

Розглянемо детальніше для подальшого дослідження традиційні та розрахункові методи.

Величину артеріального тиску прийнято розглядати як гомеостатичний показник, у зв'язку з чим його відхилення в той або інший бік може свідчити про певні зміни в загальному функціональному стані організму [14]. При його дослідженні вимірюється тиск на ліктьовій артерії з використанням тонометру або сфігмоманометру. Манжет накладається на оголене плече та накачується повітрям. Наступним етапом повільно випускається повітря та реєструються пікові тони. Поява звуків відповідає систолічному артеріальному тиску, який характеризується силою серцевого скорочення, а зникнення звуків – це діастолічний артеріальний тиск, що відповідає рівню тонусу судин. Різниця між цими показниками називається пульсовим тиском, який розраховується як різниця між величинами артеріального тиску систоли і діастоли. Враховуючи вікові особливості людини, спосіб її життєдіяльності та інші фактори, потрібно регулярно відслідковувати рівень артеріального тиску [15].

Найбільш простим у вимірюванні та інформативним критерієм стану серцево-судинної системи є частота серцевих скорочень.

Частота серцевих скорочень – це показник, який описує кількість ударів серця, тобто серцевих скорочень, за одну хвилину. В свою чергу, пульс – це

число, яке характеризує кількість разів розширення та скорочення артерій у відповідь на роботу серця за хвилину.

Частота серцевих скорочень може змінюватися залежно від фізичних потреб організму, включаючи потребу в поглинанні кисню та виведенні вуглекислого газу у відповідь. Також частота серцевих скорочень залежить від багатьох факторів, включаючи, зокрема, генетичні фактори, рівень фізичної форми та стресу або психологічного стану в цілому, дієти, вживаних ліків, гормонального фону, стану забрудненості навколишнього середовища, а також включає взаємодію між цими факторами. Зазвичай, він дорівнює або близький до пульсу, виміряного в будь-якій периферійній точці [16].

Показник частоти серцевих скорочень у нормі може різнитися від людини до людини, а також залежно від стану організму. Пульс нижчає, коли людина перебуває в стані спокою, і частішає під час фізичних навантажень, оскільки організму для цього потрібно більше збагаченої киснем крові.

Нормальною частотою серцевих скорочень у стані спокою у дорослих, тобто в осіб віком від вісімнадцяти років, вважається 60-100 ударів за хвилину, а для дітей – від 70 до 100 ударів за хвилину [15].

Якщо кількість ударів у хвилину сягає понад 100 ударів, то це може свідчити про наявність у людини тахікардії. Виділяють два її типи: фізіологічний, який може виникати через стрес, переляк чи понаднормову фізичну активність, і патологічний, який провокують певні захворювання. Якщо частота серцевих скорочень у хвилину становить менше 60 ударів, то це свідчить про наявність у людини брадикардія. Найчастіше причиною такого відхилення від норми є дисфункція синусового вузла (ділянки серця, у якій продукуються електричні імпульси).

Коли людина спить, серцебиття з частотою близько 40-50 ударів на хвилину є звичайним і вважається нормальним. Якщо серце б'ється нерівномірно, це є свідченням наявності аритмії. Порушення частоти серцевих скорочень іноді свідчать про різні захворювання.

Розглянемо фактори, які впливають на частоту серцевих скорочень. У таблиці 1.2 наведено перелік основних факторів, що збільшують частоту серцевих скорочень, а у таблиці 1.3 – фактори, які знижують частоту серцевих скорочень.

Таблиця 1.2

Фактори, що впливають на збільшення частоти серцевих скорочень

Фактор впливу	Очікуваний ефект
Кардіоакселераторні нерви	Вивільнення норадреналіну
Пропріорецептори	Збільшення темпів серцевих скорочень під час вправ
Хеморецептори	Зниження рівня O_2 ; підвищення рівня H^+ , CO_2 і молочної кислоти
Барорецептори	Зниження темпу серцевих скорочень, що вказує на падіння об'єму/тиску крові
Лімбічна система	Наслідок фізичних вправ або сильних емоцій
Катехоламіни	Підвищення адреналіну і норадреналіну
Гормони щитовидної залози	Підвищення рівню гормонів щитовидної залози
Кальцій	Підвищений рівень Ca^{2+}
Калій	Знижений рівень K^+
Натрій	Знижений рівень Na^+
Температура тіла	Підвищення температури тіла
Нікотин і кофеїн	Стимулятори, що підвищують частоту серцевих скорочень

Таблиця 1.3.

Фактори, що впливають на зниження частоти серцевих скорочень

Фактор	Ефект
Кардіоінгібіторні нерви (блукаючий нерв)	Вивільнення ацетилхоліну
Пропріорецептори	Зниження темпів частоти серцевих скорочень після вправ
Хеморецептори	Підвищення рівня O_2 ; зниження рівня H^+ і CO_2
Барорецептори	Збільшення темпів частоти серцевих скорочень, що вказує на вищий об'єм/тиск крові
Лімбічна система	Передчуття розслаблення

Фактор	Ефект
Катехоламіни	Зниження адреналіну і норадреналіну
Гормони щитовидної залози	Зниження гормонів щитовидної залози
Кальцій	Знижений рівень Ca^{2+}
Калій	Підвищений рівень K^{+}
Натрій	Підвищений рівень Na^{+}
Температура тіла	Зниження температури тіла

Частоту серцевих скорочень також вимірюють шляхом визначення пульсу серця. Цю частоту пульсу можна знайти в будь-якій точці тіла, де пульсація артерії передається на поверхню, натискаючи на неї вказівним і середнім пальцями. Великий палець не можна використовувати для вимірювання пульсу іншої людини, оскільки його сильний пульс може перешкодити правильному сприйняттю цільового пульсу.

Для перевірки частоти серцевих скорочень найпростіше використовувати променеву артерію, проте в екстрених випадках найнадійнішими є сонні артерії [17; 18]. Для вимірювання частоти серцевих скорочень також можна використовувати ліктьову артерію, внутрішню сторону ліктя або під двоголовим м'язом (плечову артерію), зону паху (стегнову артерію), зворотну частину медіальної кісточки на стопах (задню великогомілкову артерію), середину тильної поверхні стопи, підколінну артерію, зону над животом (черевну аорту), грудну клітку, зону шії, збоку від гортані (сонну артерію), зону скроні (поверхневу скронева артерію), бічний край нижньої щелепи (лицеву артерію), бічну частину голови біля вуха (задню вушну артерію).

Більш точний метод визначення частоти серцевих скорочень передбачає використання електрокардіографа або інших пристроїв для оптичного вимірювання.

Максимальна частота серцевих скорочень — це пов'язана з віком найбільша кількість ударів серця за хвилину при досягненні точки виснаження без серйозних проблем через фізичне навантаження [19]. Загалом, цей показник приблизно оцінюють як 220 мінус вік людини [20]. Зазвичай, цей

показник зменшується з віком. Досі вважається, що найточнішим способом вимірювання максимальної частоти серцевих скорочень будь-якої окремої людини є серцевий стрес-тест. У цьому тесті людина піддається контрольованому фізіологічному стресу, зазвичай, за допомогою бігової доріжки, та з урахуванням роботи електрокардіограми. Інтенсивність вправи періодично збільшують, доки на моніторі електрокардіографа не будуть виявлені певні зміни в роботі серця та дихальної системи, після чого експеримент припиняється. Типова тривалість тесту становить від десяти до двадцяти хвилин [21].

Теоретична максимальна частота серцевих скорочень людини становить 300 уд./хв., однак, було зафіксовано багато випадків, коли ця теоретична верхня межа була перевищена. Найшвидша швидкість шлуночкової провідності людини, яка була зареєстрована – це проведена тахіаритмія з частотою шлуночків 480 ударів на хвилину [22], що можна порівняти з частотою серцевих скорочень миші.

У дослідженні 2002 року, опублікованому в Journal of Exercise Psychology [23], де було розглянуто сорок три різні формули для максимального значення частоти серцевих скорочень, прийшло до наступних висновків:

1. До зазначеного вище етапу дослідження не існувало раніше формули, за допомогою якої можна було одночасно прорахувати діапазони частоти серцевих скорочень та врахувати всі фактори впливу на цей показник.
2. Формула (1.2) викликає найменше заперечень [24]:

$$HR_{max} = 205,8 - (0,685 \times \text{вік}) \quad (1.2)$$

Максимальна частота серцевих скорочень істотно відрізняється між людьми. Як зазначається в [22], навіть в одній елітній спортивній команді, наприклад, у 20-річних олімпійських веслярів, максимальна частота серцевих скорочень, може коливатися від 160 до 220 ударів на хвилину. Така варіація прирівнюється до 60-річної або 90-річної різниці у віці. Крім того, варто

розуміти, що люди одного віку, однакового рівня натренованості, у тому самому виді спорту, у тій самій команді можуть мати фактичну максимальну частоту серцевих скорочень, яка відрізняється на 60 ударів на хвилину один від одного, тобто

$$HR_{max} \in (160-220) \quad (1.3)$$

Вказаний вище інтервал достатньо широкий для того, щоб зробити висновок, що частота серцевих скорочень є ключовим показником діяльності серцево-судинної системи.

Ще одним показником функціонування серцево-судинної системи людини є варіабельність серцевого ритму. Цей показник характеризує зміни часового інтервалу між ударами серця та вимірюється зміною інтервалу між ударами. Варіабельність серцевого ритму – це не одне число, адже ця метрика може охоплювати групу тестів на основі часу та частоти, кожен із яких має власний розрахунок. Вимірювання варіабельності серцевого ритму дає людині змогу контролювати гнучкість нервової системи.

Варіабельність серцевого ритму має зовнішнє і внутрішнє походження. До зовнішніх причин відносять зміну тіла у просторі, фізичне навантаження, психоемоційний стрес, а також температуру навколишнього середовища [25]. В стані спокою серце скорочується практично з постійною частотою. Як зазначалося вище, лабільність частоти серцевих скорочень зумовлюється вегетативними впливами на синусний вузол. Симпатичні імпульси прискорюють ритм серця, а парасимпатичні його сповільнюють. Основна мета регуляції частоти серцевих скорочень – стабілізація артеріального тиску. Крім вегетативних впливів на серце, зміни в частоті серцевих скорочень викликають і гуморальні фактори. Коливанням концентрації у крові адреналіну та інших гуморальних агентів пояснюють походження дуже повільних хвиль серцевого ритму. Також зміни цього показника при диханні пов'язують із функціонуванням барорефлекторної системи стабілізації артеріального тиску. Зміни руху грудної клітки і діафрагми при диханні

призводять до коливань тиску в грудній порожнині, що проявляє збуджувальний вплив на систему стабілізації артеріального тиску. Як відомо, серцевий викид зменшується на вдиху і збільшується на видиху, внаслідок зміни припливу крові до серця та при зміні тиску в грудній порожнині. Тому, можна вважати, що особливий вплив на варіабельність серцевого ритму має легенева вентиляція, зокрема, стимуляція хеморецепторів викликає помірну гіпервентиляцію, з боку серця при цьому виявляють брадикардію, і, навпаки, при значній гіпервентиляції частота серцевих скорочень, зазвичай зростає, що має вплив на зміну показників варіабельності серцевого ритму [25; 26].

Варіабельність серцевого ритму у здорових людей дозволяє оцінити їхні фізіологічні показники, які визначаються статтю, віком, розміщенням тіла в просторі, температурою навколишнього середовища, психологічним комфортом, часом доби, сезонністю та низкою інших факторів. Дані показники відрізняються високим ступенем індивідуальності, а про порушення регуляції йдеться тоді, коли показники виходять за межі значень індивідуальної норми. Проте, різниці за статтю у показниках варіабельності серцевого ритму немає, хоча у жінок частота серцевих скорочень вища, ніж у чоловіків. В цілому, з віком пов'язують зниження загальної потужності спектру варіабельності серцевого ритму за рахунок переважного зниження низькочастотного на високочастотного компонентів. Найбільш висока потужність спектру спостерігається у дитячому і юнацькому віці. З віком реакція на модуляцію дихання знижується, що пов'язують з втратою фізичної активності. Маса тіла також впливає на варіабельність серцевого ритму. Менша маса тіла проявляється більш високою потужністю спектра високочастотного компонента, а у людей з надмірною масою тіла відзначають зворотну залежність. Добові коливання варіабельності серцевого ритму проявляються більшою потужністю спектра низькочастотного компоненту вдень і меншою вночі, при чому максимальні показники можна зафіксувати, провівши вимірювання вранці.

Фізичні вправи і спорт сприяють позитивним змінам варіабельності серцевого ритму, зокрема, при регулярній фізичній активності знижується частота серцевих скорочень, а потужність спектру варіабельності серцевого ритму зростає за рахунок високочастотного компоненту. Слід пам'ятати, що надмірні тренування можуть супроводжуватись підвищенням пульсу і, відповідно, зниженням варіабельності серцевого ритму. Цим частково і пояснюються випадки раптової смерті, пов'язаної з надмірними навантаженнями, що найчастіше проявляється у професійному спорті.

Одна із проблем дослідження варіабельності серцевого ритму полягає в тому, що всі параметри, які використовуються для характеристики цього показника, сильно залежать від частоти серцевих скорочень, але багато статей не враховували повною мірою різницю частоти серцевих скорочень при порівнянні варіабельності серцевого ритму за багатьох обставин. Саме тому точне співвідношення показників частоти серцевих скорочень та варіабельності серцевого ритму є досі до кінця не дослідженим [27]. Для параметрів часової області результати досліджень вказують на те, що, якщо існує універсальна функція, яка описує взаємозв'язок цих двох показників, то вона має бути або експоненціальною, або гіперболічною за своєю природою, проте, однозначної відповіді ще не можна вказати [28; 29].

Серцевий індекс – це розрахунковий показник, який характеризує гемодинамічні можливості організму. В цілому, у фізично здорових людей величина серцевого індексу залежить лише від двох параметрів – стать і вік. Послаблення можливостей функціонування серця з перекачування крові внаслідок захворювань призводить до зниження серцевого індексу, але це вже індивідуалізовані випадки.

Серцевий індекс не вимірюються конкретним приладом, тому для його визначення необхідно знати інші величини. Серед величин, які дозволяють визначити серцевий індекс та які мають вплив на його значення можна виділити обсяг кровообігу в одну хвилину (хвилиний обсяг кровообігу), та загальну площу поверхні тіла досліджуваної людини.

Хвилиний обсяг кровообігу – це кількість крові, яка перекачується обома відділами серця за одну хвилину в серцево-судинній системі. Даний показник вважається одним із найважливіших для визначення функціонального стану організму. Розмірність показника хвилиного обсягу кровообігу – літри/хвилину [8; 30].

Хвилиний обсяг кровообігу розраховується за формулою (1.3)

$$\text{ХОК} = \text{УО} \times \text{ЧСС} \quad (1.3)$$

де

ХОК – хвилиний обсяг кровообігу,

УО – ударний об'єм серця, який вимірюється в літрах,

ЧСС – частота серцевих скорочень (уд./хв.).

Також хвилиний обсяг кровообігу можна визначати за допомогою інших методів. Одним із таких методів дослідження є використання реєстрації розведення і «зігрівання» введеного фізіологічного розчину або глюкози кімнатної температури до внутрішньої температури в кров'яному руслі. Для проведення коректного дослідження достатньо 5-10 мл розчину. Відповідне програмне забезпечення в стані зареєструвати і швидко обчислити параметри, які необхідні для аналізу. При проведенні цього дослідження слід точно дотримуватися вимог проведення вимірювання, оскільки порушення призводить до неточних результатів.

1. Швидке введення розчину (протягом чотирьох секунд).
2. Момент введення розчину повинен збігатися з максимальним видихом.
3. Провести два вимірювання і взяти середній показник, при цьому різниця не повинна перевищувати 10%.

Нормальними вважаються коливання хвилиного обсягу кровообігу в межах 4-6 л./хв. Аналіз цього показника серцево-судинної діяльності людини дає можливість провести аналіз функціональних можливостей серцево-судинної системи і гемодинамічного резерву серця. При великих фізичних

навантаженнях хвилиний обсяг кровообігу у здорової людини досягає 25-30 л./хв., а у людей, що активно займаються спортом може сягати до 30-40 л./хв. [31]

Для розрахунку площі загальної поверхні людського тіла, використовують формулу Дюбуа, в якій використовують скориговані коефіцієнти ваги тіла в кілограмах та зросту людини в метрах [32]. Формула (1.4) задає загальний вигляд формули площі тіла:

$$S = (M \times 0,423) \times (Z \times 0,725) \times 0,007184 \quad (1.4.)$$

де

M – маса тіла (кг),

Z – зріст (м),

0,007184 – сталий коефіцієнт.

Показник серцевого індексу дозволяє нівелювати відмінності пацієнтів за масою і зростом та враховувати залежність тільки від хвилинного кровотоку. Тому цей показник може підвищуватися при збільшенні викиду в наступних випадку наявності гіпоксії тканин міокарда, підвищеному вмісті вуглекислоти в крові, тахікардії, підвищеній температурі тіла, прискореному метаболізмі, стресовому стані, в початковій стадії шоку тощо.

Зменшення серцевого індексу можуть супроводжувати:

1. Шоковий стан в третій і вищій стадії.
2. Тахікардія понад 150 ударів за хвилину.
3. Глибокий наркоз.
4. Зниження температури тіла.
5. Велика гостра крововтрата.
6. Зниження рідкої частини крові.

У здоровому організмі коливання значення серцевого індексу, як зазначалося вище, можливі за рахунок вікових особливостей та статі.

Серцевий індекс дозволяє отримати точнішу діагностичну інформацію. Важливо також мати на увазі, що цей показник будь-коли оцінюється

самостійно. Він входить до групи гемодинамічних величин спільно з артеріальним тиском, показником насиченням крові киснем, ударними індексами роботи кожного шлуночка, показником периферичного опору, а також коефіцієнтами доставки та утилізації кисню. Сучасне обладнання не потребує ручних розрахунків, а видає комплексний результат аналізу. Спеціаліст порівнює його зі стандартними нормативами або статистичними даними кожного пацієнта, проводить співвідношення з іншими аналітичними даними і судить про розмір компенсаторних можливостей або патологічних змін.

1.3. Методи аналізу варіабельності серцевого ритму

Для дослідження варіабельності серцевого ритму використовують низку методів. Методи, що використовуються для виявлення частоти серцевих скорочень та варіабельності серцевого ритму, включають в себе проведення електрокардіограми, регулярне вимірювання артеріального тиску, проведення балістокардіограми та вимірювання сигналу пульсової хвилі, який можна отримати з фотоплетизмографа. При будь-якому виборі методу дослідження важливим є оптичний або електричний метод запису цифрових сигналів. Обидва типи сигналів можуть надавати однакові базові дані про частоту серцевих скорочень за допомогою повністю автоматизованих алгоритмів для вимірювання частоти серцевих скорочень [15].

Датчики фотоплетизмографії використовують технологію, засновану на зміні яскравості світла, для вимірювання об'єму крові, який контролюється насосною дією серця. Фотоплетизмографія дозволяє неінвазивно реєструвати зміни кровотоку тканин. Можливість вимірювання варіабельності серцевого ритму за допомогою фотоплетизмографії обумовлена реєстрацією змін яскравості кольору покривних тканин за умови інтенсивного освітлення при скороченнях серця. За амплітудно-часовою характеристикою коливань сигналу, реєстрація якого відбувається, можна визначати інтервали часу між серцевими скороченнями, тобто інтервали варіабельності серцевого ритму.

Електрокардіограма вважається уніфікованим методом вимірювання варіабельності серцевого ритму, оскільки забезпечує пряме відображення електричної активності серця [15]. Датчики електрокардіографії вимірюють біопотенціал, що утворюється електричними сигналами, які контролюють розширення та скорочення камер серця. Зміни в інтервалі між ударами є фізіологічним явищем. Основними вхідними факторами є симпатична та парасимпатична нервова система і гуморальні фактори. Дихання викликає хвилі серцевого ритму, опосередковані головним чином через парасимпатичну нервову систему, і вважається, що відставання в петлі зворотного зв'язку барорецепторів може призвести до десяти-секундних хвиль серцевого ритму, які пов'язані зі зміною показника артеріального тиску.

Факторами впливу на результативність вимірювання є барорефлекс, терморегуляція, гормональний фон, цикли сну, якість харчування, фізична активність та рівень стресу.

Зниження активності парасимпатичної нервової системи або підвищення активності симпатичної нервової системи призводить до зниження показника варіабельності серцевого ритму.

Дослідження варіабельності серцевого ритму полягає у вимірюванні часових інтервалів між зубцями електрокардіограми (Рис. 1.3) і побудові на основі отриманих даних кардіоритмограми з наступним її аналізом різними математичними методами [33].

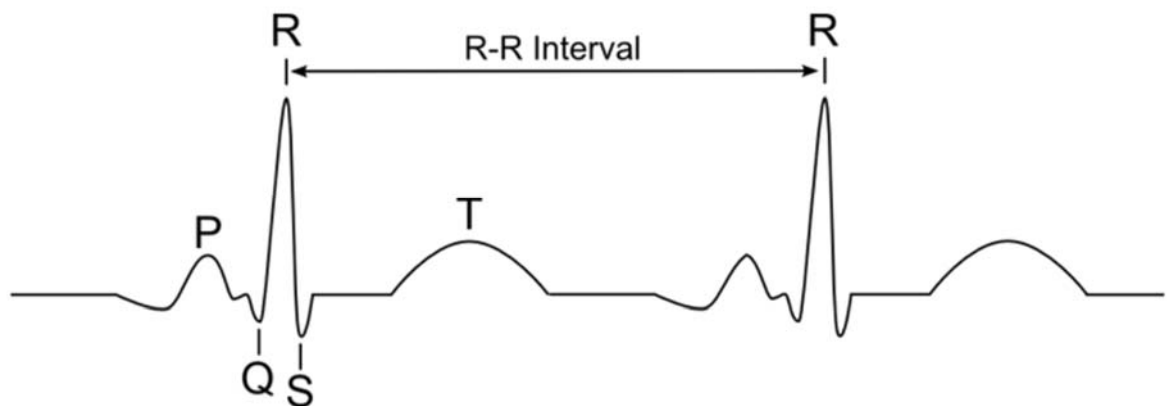


Рис. 1.3. Зображення $R - R$ інтервалу на електрокардіограмі

Нормальна форма досліджуваної хвилі, як показано на рис. 1.3, складається з кількох комплексів, відомих як хвилі P, Q, R, S і T . Комплекси позначають різні стадії серцевого циклу. Основний пік електрокардіограми, тобто комплекс $Q - R - S$, представляє деполяризацію шлуночків, що також відповідає їхньому скороченню, у той час як параметр T представляє реполяризацію і розслаблення [34].

Оскільки важливою ціллю аналізу варіабельності серцевого ритму є дослідження впливу вегетативної активності у синусовому вузлі, тому для подальшого дослідження найбільш актуальною точкою у кардіограмі є точка P . Однак, опорну точку, що зв'язана з зубцем P , надзвичайно важко визначити з достатньою точністю, оскільки зубець P має низьку амплітуду, а інколи він взагалі буває відсутнім. Відповідно, опорна точка, зазвичай, пов'язана з комплексом $Q - R - S$. Використання інтервалів $R - R$ замість інтервалів $P - P$ з часом було визнано за норму, оскільки інтервал $P - R$ можна вважати відносно фіксованим, і, таким чином, $R - R$ інтервали відображають активність синуатріального вузла [33].

Для запису кардіоритмограми, в цілому, може використовуватися запис будь-якої пульсової хвилі – реограма, сфигмограма, фотоплетизмограма та ін.

Спільна європейська та американська робоча група у 1996 році описала стандарти вимірювання варіабельності серцевого ритму [35]. До цих стандартів проведення замірювань відносять ряд вимог:

1. Тривалість короткого запису повинна складати 5 хвилин або від 300 до 500 кардіоциклів.
2. Тривалість довгого запису повинна тривати добу.

Саме за таких умов дослідження отримані результати вважатимуться коректними.

Для дослідження варіабельності серцевого ритму виділяють різні методи, серед яких виділяють статистичні, геометричні та нелінійні методи.

1.3.1. Статистичні методи

Статистичні методи застосовуються для безпосередньої кількісної оцінки варіабельності серцевого ритму в досліджуваній проміжок часу [15; 36]. При їх використанні кардіоритмограма розглядається як сукупність послідовних тимчасових проміжків, тобто інтервалів $R - R$ (рис.1.4).

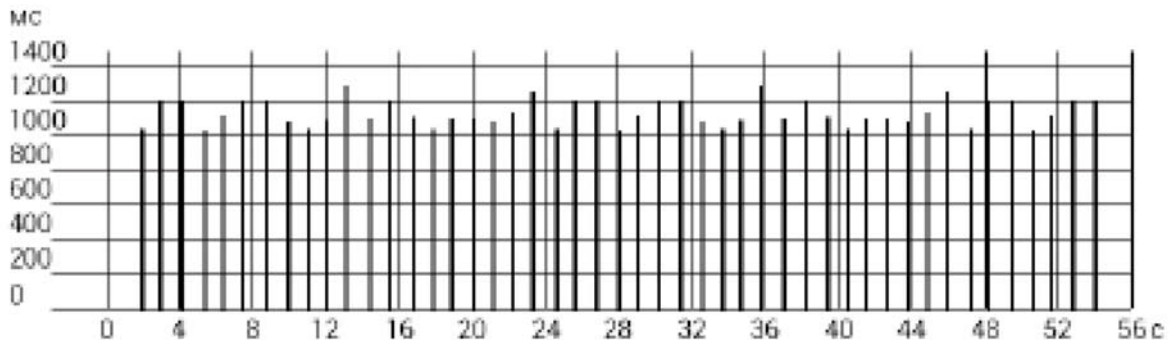


Рис. 1.4. Приклад загального вигляду кардіоритмограми

Для проведення статистичного аналізу варіабельності серцевого ритму необхідно із запису електрокардіограми виділити усі послідовні $Q - R - S$ комплекси, виміряти $R - R$ інтервали та розрахувати за ними нормальні $N - N$ інтервали, тобто інтервали між прилеглими $Q - R - S$ комплексами, які є результатом деполяризації клітин синусового вузла з виключенням екстрасистол [37]. На основі тривалих добових записів можуть бути розраховані статистичні часові показники. При аналізі відповідної кардіоритмограми методами часової області фіксують тривалість $N - N$ інтервалів та різницю між тривалістю сусідніх $N - N$ інтервалів. Кожен відрізок на кардіоритмограмі показує тривалість відповідного $R - R$ інтервалу. Дані інтервали можна фіксувати за весь період запису або за певні проміжки життєдіяльності людини, наприклад, тривалість робочого процесу, сну, періодів емоційного збудження чи стресу.

Статистичні характеристики динамічного ряду кардіоінтервалів включають в себе показники, які утворюють сумарний показник

варіабельності величин інтервалів $R - R$ за весь період аналізу. Для такого аналізу можна обчислювати наступні показники [15; 36]:

1. SDANN – стандартне відхилення середніх значень інтервалів $N - N$ з 5-тихвилинних сегментів для середньої тривалості або добових записів. Цей показник розраховується як квадратний корінь від розсіювання $N - N$ інтервалу, тобто як дисперсія [35;37]. Оскільки, дисперсія математично рівна сумарній потужності спектрального аналізу, SDNN відображає всі циклічні компоненти, що відповідають за мінливість в періоді запису [37]. Аналогічним чином можуть позначатися і стандартні відхилення середніх значень інших показників.

2. RMSSD – обчислюється як квадратний корінь з суми квадратів різниці величин послідовних пар інтервалів $N - N$. Даний параметр використовують щоб підкреслити високочастотний вміст серії $N - N$ інтервалів. Таким чином, RMSSD є часто використовуваним показником дисперсії в клінічних дослідженнях.

3. NN50 – кількість випадків, в яких різниця між тривалістю послідовних $N - N$ інтервалів за весь період запису, перевищує 50 мс.

4. PNN50 (%) – відсоток послідовних $N - N$ інтервалів, різниця між якими за весь період запису, не перевищує 50 мс.

5. CV – коефіцієнт варіації, який є зручним для практичного використання, оскільки є нормованою оцінкою сумарного показнику варіабельності серцевого ритму, який може бути знайдений за формулою (1.4)

$$CV = \frac{CKO}{M} \times 100, \quad (1.4)$$

де

CKO – сумарний показник варіабельності серцевого ритму,

M – це середнє значення інтервалів $R - R$.

1.3.2. Геометричні методи (варіаційна пульсометрія)

Сутність варіаційної пульсометрії полягає у вивченні закону розподілу на кардіоінтервали як випадкових величин [15; 36]. При цьому будується гістограма, тобто крива розподілу кардіоінтервалів, і визначаються її основні характеристики такі як мода (Mo), амплітуда моди (AMo) та варіаційний розмах (BP). Мода – статистичне значення, яке найчастіше зустрічається в динамічному ряду значення кардіоінтервалу. При нормальному розподілі і високій стаціонарності досліджуваного процесу мода практично не відрізняється від математичного сподівання (M). Амплітуда моди – число кардіоінтервалів, що відповідають значенням моди, у відношенні до обсягу вибірки. Варіаційний розмах, в свою чергу, відображає ступінь варіативності значень кардіоінтервалів в динамічному ряду, який досліджується. Він обчислюється як різниця максимального (Max) і мінімального (Min) значень кардіоінтервалів, тому при аритміях або артефактах може давати спотворене значення.

За даними варіаційної пульсометрії обчислюється індекс напруги регуляторних систем, або його ще називають стрес-індекс:

$$I_n = \frac{AMo}{2Mo} \times BP, \quad (1.5)$$

Також дослідники серед геометричних методів використовують апроксимацію кривої, тобто наближення розподілу кардіоінтервалів методом трикутників і обчислюють так званий триангулярний індекс, тобто інтеграл щільності розподілу, який відноситься до максимуму щільності розподілу (AMo).

1.3.3. Автокореляційний аналіз

Для вивчення внутрішньої структури динамічного ряду кардіоінтервалів як випадкового процесу використовують методи побудови автокореляційної функції даного ряду та обчислення її ключових показників [15; 36].

Автокореляційна функція задається як графік динамічної зміни коефіцієнта кореляції, який отримується при послідовному зсуві аналізованого динамічного ряду на одне число по відношенню до свого власного ряду.

1.3.4. Кореляційна ритмографія – скаттерографія

Суть даного методу кореляційної ритмографії полягає у графічному зображенні послідовних пар кардіоінтервалів (попереднього та наступного) у двомірній декартовій системі координат [15; 36]. При цьому на осі абсцис відкладається величина $R - R_n$, а на осі ординат - величина $R - R_{n-1}$. Графік та множина точок, отриманих таким чином, називають кореляційною ритмограмою або скаттерограмою. Нормальна форма скаттерограми – еліпс, витягнутий уздовж його осей. Цей спосіб оцінки варіабельності серцевого ритму відносять до методів нелінійного аналізу і особливо корисний він для використання у випадках, коли на тлі монотонності серцевого ритму зустрічаються рідкісні і раптові порушення.

Для аналізу з використанням кореляційної ритмографії доцільно використовувати наступні показники скаттерограми:

1. L – це довжина основної (без екстрасистол і артефактів) осі, яка відповідає варіаційному розмаху. За фізіологічним змістом цей показник не відрізняється від SADNN зі статистичного методу, тобто відображає сумарний ефект регуляції варіабельності серцевого ритму, але вказує на максимальну амплітуду коливань тривалості інтервалів $R - R$.

2. w – ширина скаттерограми (перпендикуляр до довгої осі, проведений через її середину).

3. Площа скаттерограми обчислюється за формулою площі еліпса

$$S = \frac{(\pi \times L \times w)}{4} \quad (1.6)$$

1.3.5. Інші методи аналізу варіабельності серцевого ритму

Варіабельність серцевого ритму, зазвичай, визначають кількісно, використовуючи методи часової та частотної областей [15]. Традиційні методи аналізу даних, описані вище, у часовій та частотній областях часто недостатні для характеристики складної динаміки генерації серцебиття. Це пояснюється тим, що механізми, які беруть участь у серцево-судинній регуляції взаємодіють між собою нелінійно.

Методи цифрової фільтрації призначені для швидкого аналізу короткотривалих ділянок запису електрокардіограми та дозволяють дати кількісну оцінку періодичних компонентів варіабельності серцевого ритму.

Різноманітні впливи на варіабельність серцевого ритму, включаючи нейрогуморальні механізми вищих вегетативних центрів, обумовлюють нелінійний характер змін серцевого ритму, для опису яких потрібно використання спеціальних методів. Для опису нелінійних властивостей варіабельності застосовують кластерний спектральний аналіз, сингулярне розкладання, експоненту Ляпунова, ентропію Колмогорова та інші методи нелінійного динамічного аналізу.

Застосування спектрального аналізу дозволяє кількісно оцінити частотні складові коливань ритму серця і наочно графічно представити співвідношення різних компонентів серцевого ритму, які відображають активність певних ланок регуляторного механізму. При спектральному аналізі варіабельності серцевого ритму велике значення має обсяг аналізованої вибірки. При короткотривалих записах виділяють три головних спектральних компонента, відповідні діапазони дихальних хвиль і повільних хвиль першого та другого порядків.

Всі ці методи в даний час представляють лише дослідницький інтерес, їх практичне застосування обмежене.

На поточному етапі практичного використання методів аналізу варіабельності серцевого ритму в прикладній фізіології та клінічній медицині

представлені вище підходи до практичного інтерпретування даних які дозволяють ефективно вирішувати завдання діагностичного та прогностичного характеру, оцінки функціональних проб, контролю ефективності лікувально-профілактичних впливів тощо.

1.4. Загальна концепція архітектури та формалізація вимог до програмного забезпечення

На основі проведеного огляду показників роботи серцево-судинної системи та методів їх визначення можна попередньо окреслити вимоги до програмного забезпечення. Необхідно реалізувати додаток для отримання деяких показників роботи серцево-судинної системи, таких як, варіабельність серцевого ритму та частота серцевих скорочень. Програмне забезпечення повинно отримувати на вході множину відеофайлів із записом пульсації капілярів у стінках пальців, знятих за допомогою камери смартфона. На виході, після аналізу відеофайлів, необхідно відобразити сигнал у графічному вигляді та показники роботи серця.

Для систематизації функціональних вимог відобразимо їх у вигляді UML-діаграми прецедентів.

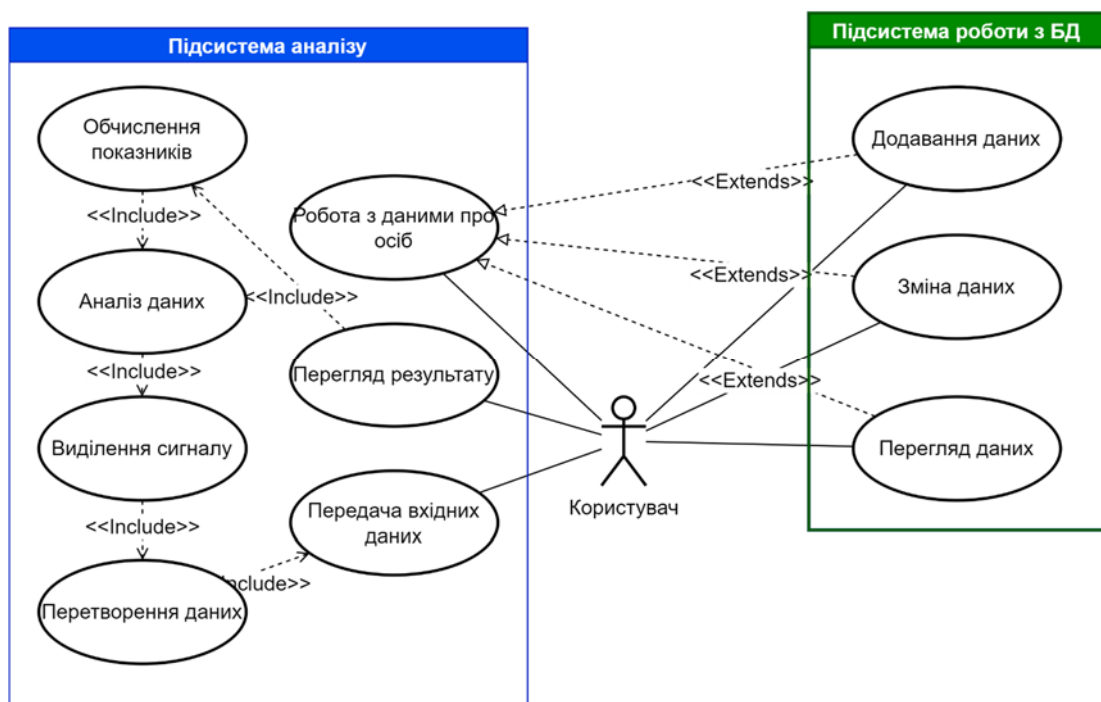


Рис.1.5 Діаграма прецедентів

Як видно, функціональність програми може бути розділена за декількома підсистемами:

1. Підсистема аналізу включає в себе функції обробки вхідних даних, обчислення показників роботи серцево-судинної системи та візуалізацію результату.

2. Підсистема обробки даних охоплює функції для створення та маніпулювання даними користувачів.

Крім того, програма повинна мати графічний інтерфейс для зручнішої роботи користувача.

Спираючись на дані вимоги доцільно виконати декомпозицію підсистеми аналізу за функціональним призначенням модулів на підсистеми препроцесингу відео, обробки сигналу та візуалізації.

Таким чином концептуально архітектуру додатку можна описати наступним чином:

1. Підсистема інтерфейсу користувача – програмні модулі для реалізації користувацького інтерфейсу.
2. Підсистема препроцесингу відео – здійснює попереднє перетворення відеопотоку, зокрема обробку кадрів та перетворення кольірних моделей.
3. Підсистема обробки сигналу – здійснює математичну обробку вхідного сигналу та обчислює показники роботи серцево-судинної системи.
4. Підсистема візуалізації – дозволяє відобразити одержані в результаті роботи попереднього алгоритму дані у вигляді графіків.
5. База даних – забезпечує зберігання інформації про вже проведені випробовування (тестування) та необхідних параметри для роботи програми
6. Підсистема для роботи з базою даних – забезпечує меназми для взаємодії програмних модулів з базою даних.

Доцільно, щоб програма могла працювати з найбільш поширеними форматами відеофайлів. Оскільки передбачається використання камери смартфона, то логічно обрати формати, у яких виконують запис сучасні смартфони, зокрема mp4, mov, avi. На виході програма додатково має зберігати всі графічні дані у формах png та jpg.

Концептуальна архітектура програмного забезпечення та зв'язок підсистем представлені на рис.1.6.

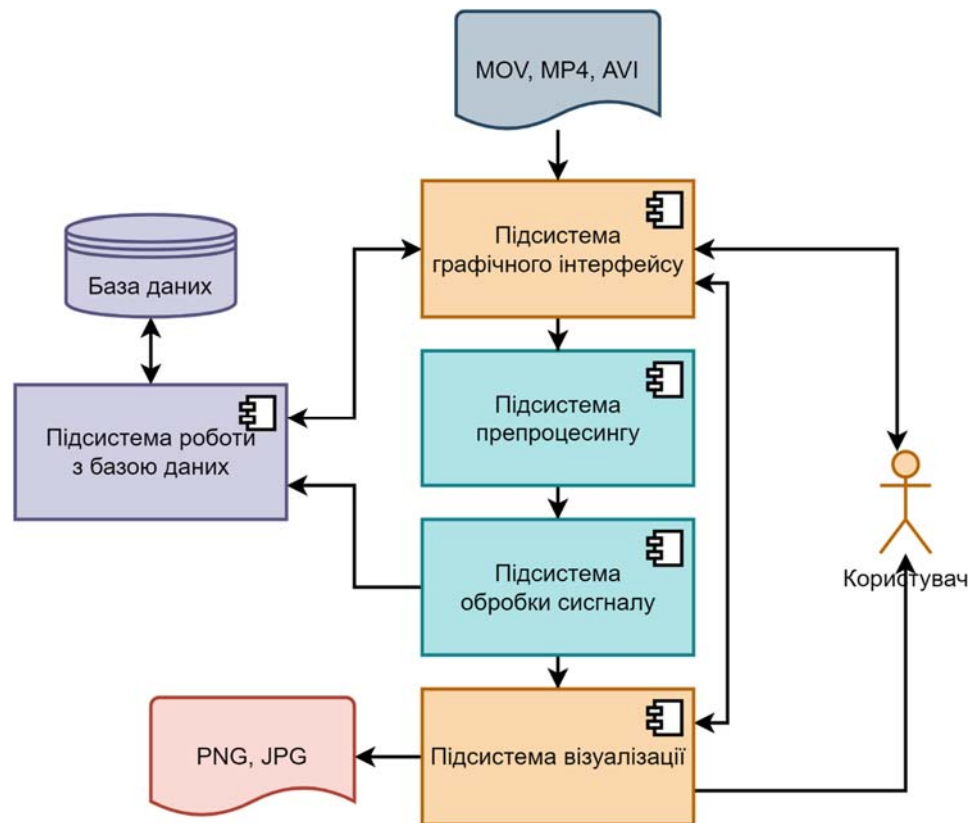


Рис.1.6. Концептуальна архітектура програмного забезпечення

Висновки до розділу 1

1. Проаналізовано сучасний стан захворюваності населення серцево-судинними захворюваннями. Встановлено основні фактори ризику та тенденції розвитку даної проблеми.
2. Встановлено та описано основні показники захворюваності серцево-судинної системи, а також методи їх визначення.
3. Проведено огляд методів аналізу варіабельності серцевого ритму.

4. Враховуючи проведений аналіз методів сформовано функціональні вимоги до програмного забезпечення.

5. Розроблено концептуальну архітектуру програмного забезпечення.

РОЗДІЛ 2

МАТЕМАТИЧНІ ТА ПРОГРАМНІ АСПЕКТИ РЕАЛІЗАЦІЇ АЛГОРИТМУ ОЦІНКИ ПОКАЗНИКІВ ДІЯЛЬНОСТІ СЕРЦЕВО- СУДИННОЇ СИСТЕМИ

2.1. Особливості використання методу фотоплетизмографії під час дослідження варіабельності серцевих скорочень з використанням камери смартфона

Вимірювання варіабельності серцевого ритму широко використовуються для оцінки функції серцево-судинної та вегетативної нервової систем. Сигнал, з якого отримуються дані, вимагає точного визначення інтервалу між послідовними ударами серця; його можна зареєструвати за допомогою електрокардіографії, яка є широко доступною. Проте методологічні проблеми, пов'язані з записом та аналізом електрокардіограми, та непрактичність щоденного діагностування спонукали до пошуку альтернатив. Як зазначалося вище, фотоплетизмографія є ще одним способом визначення часу серцевих циклів шляхом постійного моніторингу змін об'єму крові в частині периферичного мікроциркуляторного русла. Ця методика вимірює пульсові хвилі, що в багатьох випадках може бути практичною основою для аналізу варіабельності серцевого ритму та частоти серцевих скорочень. Фотоплетизмографія забезпечує точні міжпульсні інтервали, на основі яких можна точно визначити показники функціональності серцево-судинної системи у здорових суб'єктів за ідеальних умов, що свідчить про те, що цей метод може стати практичною альтернативою проведення електрокардіограми для аналізу варіабельності серцевого ритму [38].

Технологія фотоплетизмографії використовується в пульсоксиметрах, а також у багатьох фітнес-браслетах, фітнес-трекерах чи смарт-годинниках [39].

В процесі серцебиття, кров тече в капіляри пальців та витікає з них. Ці зміни в об'ємі крові змінюють колір тканини пальця. Тому, суть використання

фотоплетизмографії в такому дослідженні полягає в тому, щоб відстежувати описані вище зміни за допомогою світла або інфрачервоного випромінювання. В процесі визначення показників діяльності серцево-судинної системи шляхом реєстрації сигналу за допомогою камери телефону спалах освітлює тканину у пальці суб'єкта дослідження, а камера фіксує зміни на відео, і в результаті відбувається запис короткого відеоролику процесу серцебиття. Коли серце скорочується, відбувається викид крові та судини темніють, а коли серце розслаблюється – судини світлішають. По тому, з якою швидкістю змінюється колір, можна проаналізувати варіабельність серцевого ритму, частоту серцевих скорочень та багато інших показників.

Проте, існує ряд факторів, які здатні вплинути на точність діагностики. Для того, щоб отримати високу точність вимірювання значення варіабельності серцевого ритму за допомогою фотоплетизмографії, необхідно враховувати можливі ризики, які можуть виникнути під час процесу вимірювання. Зокрема, це якість поведінки, або що робить суб'єкт дослідження на момент процесу вимірювання та технічна якість, або оцінка якості відеозапису, яка здійснюється камерою смартфона на момент дослідження.

Якість поведінки включає положення тіла під час вимірювання, будь-які рухи, які відбуваються та, навіть, інтенсивність дихання.

Щоб забезпечити коректність результатів процесу вимірювання, досліджуваній особі слід дотримуватися наступних рекомендацій [40]:

1. Прийняти нейтральне положення. Це може бути положення лежачи або сісти, притулившись спиною до спинки стільця.
2. Перед проведенням вимірювання слід перебувати в стані спокою 10–15 хвилин задля нормалізації пульсу.
3. Під час проведення вимірювання потрібно намагатися не рухатися. Будь-який рух впливає на частоту серцевих скорочень.
4. Також не потрібно намагатися контролювати своє дихання, навмисно роблячи глибокі або рівні вдихи. Ритмічність та глибина дихання впливає на пульс, тому варто дихати природньо.

5. Не говорити, адже розмова змінює ритм дихання.

Технічна якість процесу фотоплетизмографії з використанням камери смартфона може бути низькою, якщо сигнал нечіткий або зображення розмите. В цілому, на якість отриманого відеоряду може вплинути положення пальця суб'єкта дослідження на камері та спалаху або інтенсивність зовнішнього освітлення [41]. Важливо, щоб камера смартфона знімала лише освітлений палець, тому під час проведення вимірювання потрібно повністю закрити камеру пальцем і переконатися, що спалах освітлює кровоносні судини (рис.2.1).

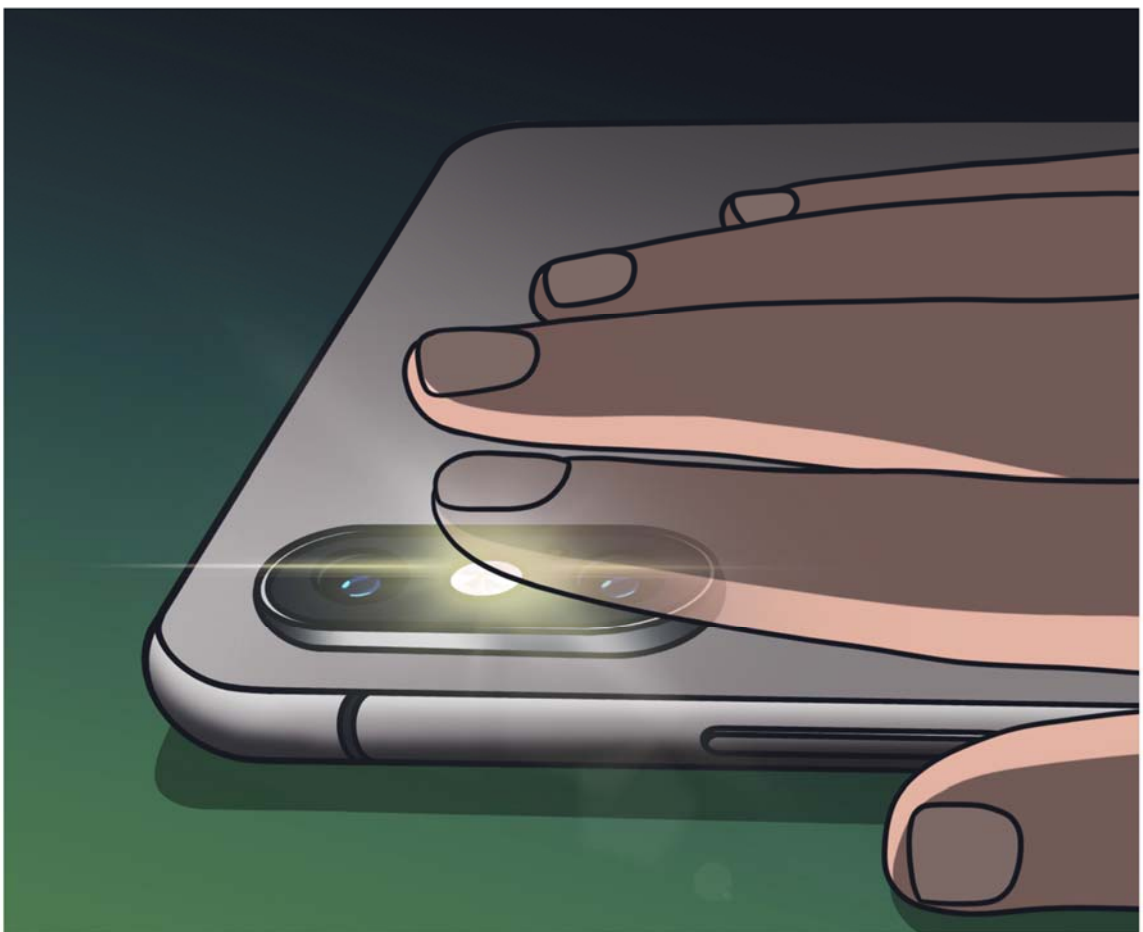


Рис. 2.1. Приклад положення пальця при проведенні вимірювання

Якщо, залежно від моделі смартфона, спалах надто нагріває шкіру пальця, то його слід тримати на відстані 2–3 мм від спалаху. Для отримання якісного сигналу важлива стабільність. Якщо під час вимірювання відбувається рух пальця або змінюється рівень його тиску на камеру, це,

імовірно, змінить колір відеоряду, який буде аналізуватися алгоритмом. В результаті таких неточностей частина серцевих ударів буде відфільтровано, що зменшить оцінку точності.

Дотримання описаних вище умов під час проведення замірів є важливою складовою при отриманні коректних результатів, оскільки будь-який сторонній фізіологічний або технічний вплив обумовлює велику кількість шумів на кардіоритмограмі і, як наслідок, неточність результатів вимірювань.

2.2. Попередній алгоритм роботи обчислювального конвеєру

Для того, щоб оптимально вибрати методи та інструментарій розробки необхідно попередньо сформулювати алгоритм роботи додатку.

Даний алгоритм покажемо у вигляді діаграми діяльності обчислювального конвеєру [42] (рис.2.2).

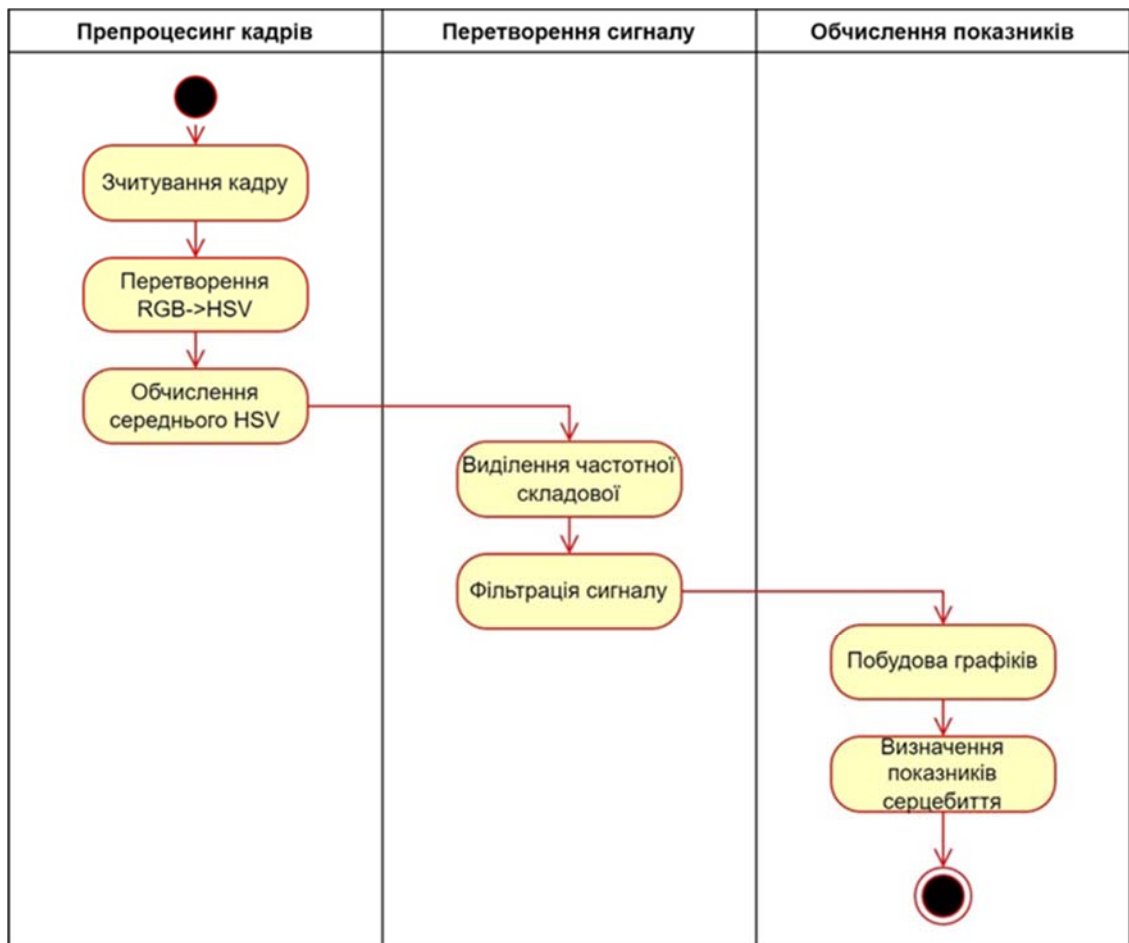


Рис.2.2. UML-діаграма діяльності обчислювального конвеєра

Діаграма охоплює не всю роботу програми, а лише ту частину, що включає в себе попередню обробку відео та визначення показників серцебиття.

Як видно з діаграми, обчислювальний конвеєр складається з трьох етапів:

1. Препроцесинг кадрів – зчитування та перетворення кадрів для подальшої роботи алгоритму.
2. Перетворення сигналу – виділення із даних, отриманих з кадрів, сигналу.
3. Обчислення показників – визначення показників серцебиття, таких як варіабельність серцевого ритму та частота серцевих скорочень, візуалізація результату.

Детальніше розглянемо кожен з етапів. Для препроцесингу кадрів необхідно зчитати кадри отриманого відеопотоку та перетворити їх. Це робиться для фіксації періодів зміни яскравості зображення при запису пульсації крові через пальці, що відповідає періодам серцевих скорочень. Саме ці перепади у яскравості кадрів будемо розглядати як вхідний сигнал.

RGB — це колірна модель із трьома параметрами — червоним, зеленим і синім, — які змішуються для створення певного кольору. Визначаючи кольори в цих межах, необхідно знати послідовність кольорів у колірному спектрі.

HSV — це циліндрична колірна модель, яка змінює основні кольори RGB на параметри, легші для розуміння людиною. Цими параметрами є відтінок, насиченість і значення. Відтінок визначає кут кольору в кольоровому колі RGB. Наприклад, відтінок 0° дає червоний колір, 120° – зелений, а 240° – синій. Насиченість контролює кількість використаного кольору. Колір зі 100% насиченістю буде найчистішим із можливих, а 0% насиченості дає відтінки сірого. Значення контролює яскравість кольору. Колір з яскравістю 0% є чисто чорним, а колір зі 100% яскравістю не містить чорного кольору. Часто цей параметр часто називають яскравістю зображення.

Оскільки модель RGB не несе інформації про яскравість зображення, то кадри потрібно перевести в модель HSV.

Дане перетворення математично можливо описати за допомогою наступних залежностей [43]:

$$\begin{aligned}
 V &= \max(R, G, B) \\
 S &= \begin{cases} \frac{V - \min(R, G, B)}{V}, & \text{якщо } V \neq 0 \\ 0, & \text{в інших випадках} \end{cases} \\
 H &= \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)}, & \text{якщо } V = R \\ \frac{120 + 60(B - R)}{V - \min(R, G, B)}, & \text{якщо } V = G, \\ \frac{240 + 60(R - G)}{V - \min(R, G, B)}, & \text{якщо } V = B \end{cases}
 \end{aligned} \tag{2.1}$$

де

R, G, B – значення інтенсивності кольору (червоного, зеленого та синього) для кожного пікселю,

H, S, V – відтінок, насиченість та яскравість відповідно.

На виході отримуємо тривимірні масиви даних із значеннями яскравості, насиченості та відтінку кольору.

Після перетворення кадрів відео в колірну модель HSV відбувається їх усереднення за яскравістю пікселів для кожного кадру.

Наступним кроком є виділення частоти зміни яскравості зображення з вихідних даних. Після перетворення сигналу і його фільтрації визначається показник варіабельності серцевого ритму та частота серцевих скорочень (на хвилину). Методи обчислення цих показників буде розглянуто нижче.

Після визначення всіх показників результати потрібно вивести в графічному вигляді та записати до бази даних.

2.3. Математична модель реалізації алгоритму вимірювання варіабельності серцевого ритму

Для подальшої роботи з даними отриманими при виділенні кадрів з відеофайлу необхідно виділити із них сигнал та провести його подальший аналіз.

В широкому розумінні, сигнал – це інформаційна функція, яка несе в собі повідомлення про фізичні властивості, стан або поведження певної фізичної системи, об'єкта або середовища [44]. Метою обробки сигналів будемо вважати отримання певних інформаційних відомостей, що відображені в цих сигналах і подальше перетворення цих відомостей у зручну для сприйняття й подальшого використання форму. Проведення математичних перетворень та одержання на основі цих перетворень висновків стосовно специфічних особливостей відповідних процесів та об'єктів є ключовою ідеєю аналізу сигналів [45]. Метою аналізу сигналів є визначення або оцінка числових параметрів сигналів (кількість енергії, середня потужність, середньоквадратичне відхилення та інші) та розкладання сигналів на елементарні частини для порівняння властивостей різних сигналів.

Для реалізації виділення частоти зміни яскравості зображення з вихідних даних, послідовність отриманих кадрів сприйматимемо як цифровий сигнал, який залежить від функції частоти. Слід зазначити, що цей сигнал буде дискретним, на що впливатимуть як $R - R$ так і $N - N$ інтервали з кардіоритмограми, отриманої при записі відео та при препроцесингу [44; 45].

Найважливішою характеристикою вихідного сигналу є його перетворення Фур'є [46]. Якщо вихідний сигнал $f(t)$, заданий на всій дійсній осі, то його перетворення Фур'є має вигляд:

$$F(if) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi f(t)} dt \quad (2.2)$$

Функція $F(if)$ або її модуль трактується як інтенсивність вихідного сигналу на частоті f . Зворотне перетворення можна задати формулою (2.3)

$$f(t) = \int_{-\infty}^{+\infty} F(if) e^{i2\pi ft} df \quad (2.3)$$

Для існування перетворення Фур'є від функції сигналу необхідне її спадання до нескінченності, що у випадку аналізу послідовності кадрів, отриманих з відеосигналу, досягається за замовчуванням [45].

Дискретне перетворення Фур'є є вибіркоким перетворенням Фур'є, тому містить не всі частоти, що формують зображення, а лише набір вибірок, який є достатньо великим, щоб повністю описати зображення просторової області. Кількість частот відповідає числу пікселів у просторовій області зображення, тобто зображення у просторовій та Фур'є-областях мають однаковий розмір. Основними функціями даного перетворення є синусоїда та косинусоїдна хвиля зі зростаючими частотами. Аналогічним чином можна виконувати повторне перетворення Фур'є. За допомогою записаних вище двох формул зображення просторової області спочатку перетворюється на проміжне зображення за допомогою n одновимірних перетворень Фур'є. Це проміжне зображення потім перетворюється на кінцеве зображення, знову ж таки за допомогою n одновимірних перетворень Фур'є. Вираз двовимірного перетворення Фур'є через серію з $2n$ одновимірних перетворень зменшує кількість необхідних обчислень. Навіть з такою економією обчислювальних ресурсів звичайне одновимірне дискретне перетворення Фур'є дає позитивні результати для стаціонарних сигналів, а його реалізація не потребує великих витрат. Проте, з ускладненням вирішуваних задач виникають обмеження застосування перетворення Фур'є, зокрема для сигналів, які втрачають стаціонарність. Отже, оскільки отриманий сигнал не можна вважати стаціонарним, результат перетворення Фур'є стає функцією не тільки частоти, але й часу. Для обчислення спектральних складових за сукупністю значень доцільно використати алгоритм швидкого дискретного перетворення Фур'є, який дозволяє перетворити сукупність тимчасових відліків в комплексний дискретний спектр [45; 46].

Для множини чисел a_0, a_1, \dots, a_{n-1} швидке перетворення дозволяє отримати множину чисел b_0, b_1, \dots, b_{n-1} , які можна знайти за формулою (2.4)

$$b_i = \sum_{j=0}^{n-1} a_j \varepsilon^{ij}, \quad (2.4)$$

де

$$\varepsilon^n = 1 \text{ і}$$

$$\varepsilon^k \neq 1 \text{ при } 0 < k < n.$$

Детально алгоритм застосування швидкого перетворення Фур'є описаний в праці Джеймса Вільяма Кулі [47].

Реалізація застосування швидкого перетворення Фур'є відбувається наступним чином [48]. Швидке перетворення Фур'є створює вихідне зображення з комплексним числом, яке можна відобразити двома зображеннями, або з дійсною та уявною частинами, або з величиною та фазою. Під час обробки зображень часто відображається лише величина перетворення Фур'є, оскільки воно містить більшу частину інформації про геометричну структуру зображення просторової області. Однак, якщо виникає необхідність повторно перетворити зображення Фур'є в правильну просторову область після деякої обробки в частотній області, необхідно забезпечити збереження як величини, так і фази зображення Фур'є.

При використанні прямого швидкого перетворення Фур'є для перетворення зображення з просторової області в частотну, найнижчі частоти часто відображаються великим піком у центрі даних. Діапазон значень від низькочастотного піку до високочастотного шуму є екстремальним, що свідчить про те, що зображення має фоновий шум. Поверхнєве представлення спектра потужності допомагає визначити поріг, необхідний для видалення шуму із зображення. Тоді стає доцільним створення маски для фільтрації шуму та обчислення зворотного швидкого перетворення Фур'є для створення чіткішого зображення [48].

Зображення в області Фур'є має набагато більший діапазон, ніж зображення в просторовій області. Для формування дискретного сигналу, таким чином, ми отримуємо ту саму частотну область, але періодичну в заданому діапазоні.

2.4. Особливості застосування програмних рішень

Додаток та всі його модулі розроблено мовою програмування високого рівня Python. Всі програмні модулі використовують стандартну бібліотеку Python, а також сторонні рішення на цій мові.

Для етапу препроцесингу відео, зокрема стадій розбиття відео на кадри, перетворення колірних моделей та визначення яскравості кадру застосовано бібліотеку машинного зору OpenCV [49].

OpenCV - це бібліотека з відкритим вихідним кодом, яка включає кілька сотень алгоритмів комп'ютерного зору. OpenCV має модульну структуру, що означає, що пакет включає в себе кілька загальних або статичних бібліотек. Доступні наступні модулі:

- Core functionality(core) - компактний модуль, що визначає основні структури даних, включаючи щільний багатовимірний масив Mat і основні функції, що використовуються всіма іншими модулями.

- Image Processing (imgproc) - модуль обробки зображень, що включає лінійну і нелінійну фільтрацію зображень, геометричні перетворення зображення (зміна розміру, афінне і перспективне викривлення, загальне табличне ремаппінг), перетворення колірного простору, гістограми і так далі.

- Video Analysis(video) - модуль аналізу відео, що включає в себе алгоритми оцінки руху, віднімання фону і відстеження об'єктів.

- Калібрування камери і 3D реконструкція (calib3d) - основні алгоритми багатовидової геометрії, калібрування одиночних і стереокамер, оцінка пози об'єкта, алгоритми стереовідповідності і елементи 3D-реконструкції.

- 2D Features Framework(features2d) - помітні детектори функцій, дескриптори та дескриптори.
- Object Detection(objdetect) - виявлення об'єктів і екземплярів заздалегідь визначених класів (наприклад, осіб, очей, людей, автомобілів і так далі).
- Графічний інтерфейс високого рівня (highgui) - простий у використанні інтерфейс користувача з простими можливостями інтерфейсу користувача.
- Video I/O(videoio) - простий у використанні інтерфейс для захоплення відео і відеокодеків.
- Деякі інші допоміжні модулі, такі як тестові обгортки FLANN і Google, обгортки Python та інші.

OpenCV обробляє всю пам'ять автоматично. Структури даних, що використовуються функціями та методами, мають деструктори, які мають справу з базовими буферами пам'яті, коли це необхідно. Це означає, що деструктори не завжди мають справу з буферами. Вони враховують можливий обмін даними. Деструктор видаляє опорний лічильник, пов'язаний з буфером матричних даних. Буфер працює тоді і тільки тоді, коли опорний лічильник досягає нуля, тобто коли ніякі інші структури не відносяться до того ж буферу. Аналогічно, коли копіюється екземпляр `mat`, фактичні дані насправді не копіюються. Замість цього посилюється лічильник посилянь, щоб запам'ятати, що є інший власник тих же даних.

Багато функцій OpenCV обробляють щільні 2-вимірні або багатовимірні числові масиви. Зазвичай такі функції приймають як параметри. Щоб уникнути багатьох дублікатів в API, були введені спеціальні класи «проксі». Базовим класом «проксі» є `cv::InputArray`. Він використовується для передачі масивів, доступних лише для читання, на функціональному вході. Похідний від `InputArray` клас `cv::OutputArray` використовується для зазначення вихідного масиву для функції.

OpenCV використовує винятки для сигналу про критичні помилки. Коли вхідні дані мають правильний формат і належать до заданого діапазону значень, але алгоритм з якихось причин не може досягти успіху (наприклад, алгоритм оптимізації не збігся), він повертає спеціальний код помилки (як правило, просто булева змінна).

Поточна реалізація OpenCV повністю підлягає повторному використанню. Тобто одна і та ж функція або одні й ті ж методи різних екземплярів класів можуть викликатися з різних потоків. Крім того, один і той же `Mat` може використовуватися в різних потоках, оскільки операції підрахунку посилань використовують атомні інструкції, специфічні для архітектури.

Для захоплення відео з відеофайлів, послідовностей зображень або камер використовується OpenCV клас `VideoCapture`. Клас надає API C++ для захоплення відео з камер або для читання відеофайлів і послідовностей зображень.

Код нижче демонструє використання класу:

```
cv::VideoCapture::VideoCapture (const String & filename,
int apiPreference
)
```

Параметри:

— `filename` — це може бути ім'я відеофайлу (напр. `video.avi`), або послідовність зображень (наприклад, яка буде зчитувати зразки `img_00.jpg`, `img_01.jpg`, `img_02.jpg`, ...), або URL-адреса відеопотоку (напр. `protocol://host:port/script_name?script_params|auth`), або рядок конвеєра `GStreamer` у форматі `gst`

— `apiPreference` — бажані сервери Capture API для використання. Може використовуватися для забезпечення певної реалізації.

Для того, щоб виділити сигнал із послідовності кадрів, перетворимо їх в колірну модель HSV. У OpenCV доступно понад 150 методів перетворення колірного простору. Для перетворення кольору ми використовуємо функцію

`cv.cvtColor(input_image, flag)`, де прапорець визначає тип перетворення. Для BGR→HSV, ми використовуємо прапор `cv.COLOR_BGR2HSV`.

Для HSV діапазон відтінків становить [0,179], діапазон насиченості - [0,255], а діапазон значень - [0,255]. Різні програми використовують різні масштаби. Так що якщо ви порівнюєте з ними значення OpenCV, вам потрібно нормалізувати ці діапазони.

У випадку 8-бітових та 16-бітових зображень R, G та B перетворюються на формат з плаваючою комою та масштабуються відповідно до діапазону від 0 до 1.

Для перетворення значення кольору кожного пікселю з колірної моделі RGB в колірну модель HSV використовуються залежності [50] (2.1).

На виході отримуємо тривимірні масиви даних із значеннями яскравості, насиченості та відтінку кольору. Після перетворення кадрів відео в колірну модель HSV відбувається їх усереднення за яскравістю пікселів для кожного кадру [51].

В додатку перетворення реалізується наступним кодом

```
while ret:
    ret, frame = cap.read()

    if not ret:
        break

    .....
    hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
    hsv_values = hsv[:, :, 2]
    .....
    mean_hsv_values = np.sum(np.sum(hsv_values,
axis=0), axis=0) / hsv_values.size

    hsv_v_values.append(mean_hsv_values)
```

У циклі відбувається прохід по всіх кадрах, які як зображення передаються на вхід функції `cvtColor`. Вона за допомогою обробника `COLOR_BGR2HSV` виконує відповідне перетворення і результат записує в

список `hsv`. Після цього функція обчислює середнє. Значення `hsv` для кадру і записує його в список

Для математичних обчислень та роботи з даними використовуються функції бібліотеки `NumPy` мови `Python`.

`NumPy` – це бібліотека `Python` з відкритим вихідним кодом, яка дозволяє ефективно маніпулювати багатовимірними числовими структурами даних [52]. Вони називаються масивами в `NumPy`.

Хоча типи даних `Python` можна використовувати для побудови багатовимірних структур даних (списків, що містять списки), `NumPy` швидший і забезпечує краще API для розробників. Списки `Python` мають загальне призначення. Вони можуть містити дані різних типів. Це означає, що типи також зберігаються, код диспетчеризації типу викликається під час виконання та типи перевіряються. Списки обробляються за допомогою циклів або представлень і не можуть бути векторизовані для підтримки поелементних операцій. `NumPy` жертвує деякою гнучкістю `Python` для підвищення продуктивності.

Зокрема, `NumPy` краще може впоратися з цими аспектами:

- розмір: структури даних `NumPy` займають менше місця. Кожен цілочисельний об'єкт `Python` займає 28 байт, тоді як у `NumPy` ціле число становить лише 8 байт.

- продуктивність: код `NumPy` працює швидше, ніж код `Python`, особливо для великих вхідних даних.

- функціональність: `NumPy` надає безліч функцій і методів для спрощення операцій. Також включені операції високого рівня, такі як лінійна алгебра.

Масиви `NumPy` однорідні, що означає, що елементи масиву однотипні. Отже, під час виконання не потрібна перевірка типу. Всі елементи масиву займають однакову кількість місця.

Інтервал між елементами вздовж осі також постійний. Це називається крокуванням. Це корисно, коли ті самі дані в пам'яті можна використовувати

для створення нового масиву без копіювання. Тому різні масиви - це різні погляди на пам'ять. Таким чином, легше змінювати підмножини даних в пам'яті.

Операції векторизуються, а значить, операція може виконуватися паралельно на декількох елементах масиву. Це прискорює обчислення.

NumPy надає APIs для зручного маніпулювання масивами. Деякі з них - індексація, нарізка, зміна форми, укладання та розщеплення.

Для реалізації дискретного перетворення Фур'є застосовано функцію `numpy.fft.fft` бібліотеки NumPy [53]. Її синтаксис наступний

```
fft.fft(a, n=None, axis=- 1, norm=None)
```

Параметри функції:

- `a` - вхідний масив, може бути комплексним.
- `n` `int`, необов'язково Довжина трансформованої осі виводу. Якщо `n` менше довжини входу, вхід обрізається. Якщо вона більше, вхід доповнюється нулями. Якщо `n` не дано, використовується довжина входу по осі, заданої змінною `axis`.
- `axis` `int`, необов'язково Вісь, над якою потрібно обчислити дискретне перетворення Фур'є. Якщо не задана, то використовується остання вісь.

Функція повертає комплексний масив, усічений або доповнений нулями, перетворений уздовж осі `axis`, або останньої, якщо `axis` не вказана.

Дана функція застосовується для обчислення вектора частот

```
fcomp = np.fft.fft(sig,n)
PSD = fcomp * np.conj(fcomp) / n
freq = (1/(dt*n)) * np.arange(n)
print(freq[0:45])
L = np.arange(1, np.floor(n/2), dtype="int")
```

Для фільтрації сигналу застосовується обернене дискретне перетворення Фур'є, функція `numpy.fft.ifft` бібліотеки NumPy [54].

Її синтаксис наступний

```
fft.ifft(a, n=None, axis=- 1, norm=None)
```

Ця функція обчислює обернене одновимірне дискретне перетворення Фур'є. Вхід функції повинен бути впорядкований:

- $a[0]$ повинен містити нульовий частотний терм,
- $a[1:n//2]$ повинні містити додатно-частотні терми,
- $a[n//2 + 1:]$ повинна містити терми з від'ємною частотою, в збільшення порядку, починаючи з найбільш від'ємної частоти.

Параметри функції:

- a - Вхідний масив, може бути комплексним.
- n int, необов'язково Довжина трансформованої осі виводу. Якщо n менше довжини входу, вхід обрізається. Якщо вона більше, вхід доповнюється нулями. Якщо n не дано, використовується довжина входу по осі, заданої змінною $axis$.
- $axis$ int, необов'язково Вісь, над якою потрібно обчислити обернене дискретне перетворення Фур'є. Якщо не дано, то використовується остання вісь.

Функція повертає комплексний масив, усічений або доповнений нулями, перетворений уздовж осі $axis$, або останньої, якщо $axis$ не вказана.

Фрагмент коду, який реалізує фільтрацію сигналу

```
sigfilt = np.real(np.fft.ifft(ffilt))
for nc in range(len(sigfilt)):
    sigfilt[nc] = sigfilt[nc].real
```

Висновки до розділу 2

1. Описано процес експериментального визначення показників серцево-судинної системи шляхом аналізу відеоданих.

2. Сформульовано та описано математичний апарат реалізації алгоритму вимірювання варіабельності серцевого ритму.
3. Розроблено попередній алгоритм роботи обчислювального конвеєру аналізу відеоряду.
4. Встановлено особливості роботи з бібліотекою OpenCV, використання якої і буде забезпечувати трансформацію кадрів, одрежаних з відеоданих.
5. Адаптовано можливості бібліотеки numpy в контексті використання алгоритму швидкого перетворення Фур'є для виділення частотної області сигналу.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ АНАЛІЗУ ВІДЕОДАНИХ

3.1. Особливості розробленого програмного забезпечення

Додаток розроблено за допомогою мови програмування високого рівня Python [55]. Він може використовуватись з графічним інтерфейсом, а також приймає параметри командного рядка. Взаємодія користувача з додатком та внутрішніх підсистем між собою показана на діаграмі послідовності.

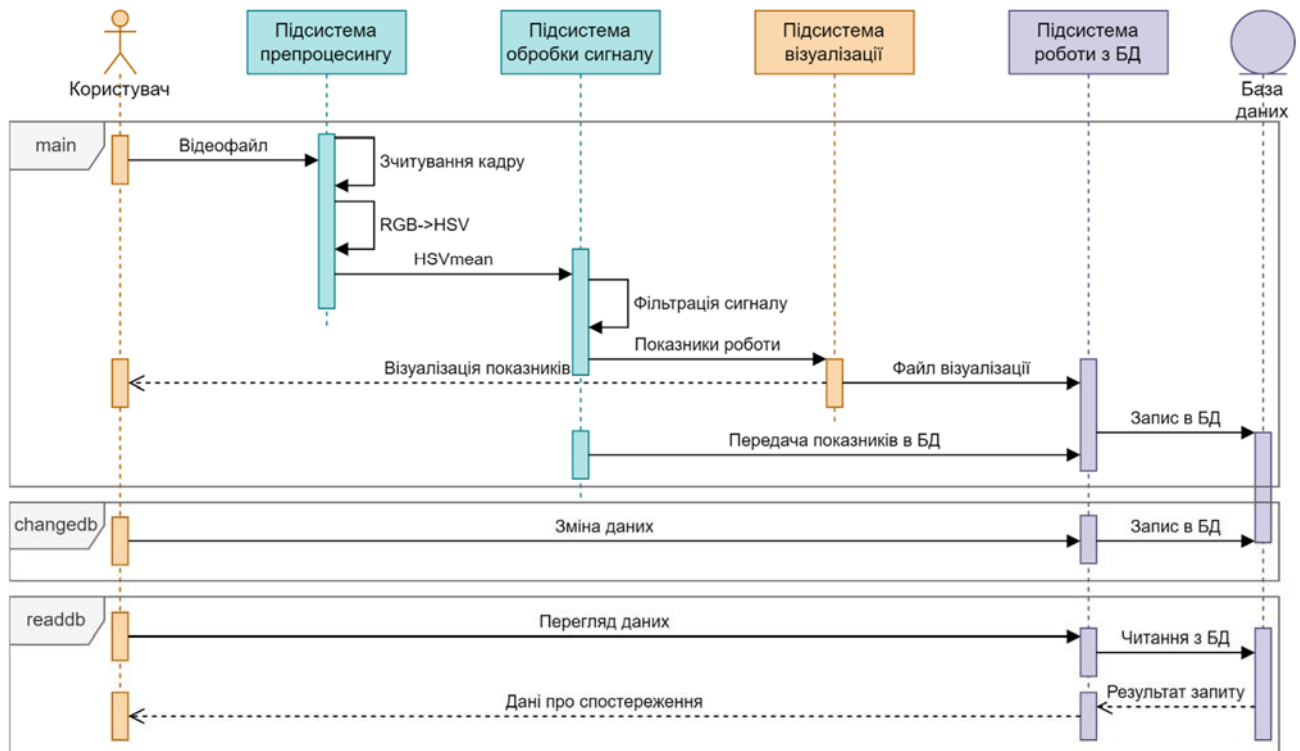


Рис.3.1. Діаграма послідовності додатку

Як видно з діаграми, в цілому взаємодія з програмою здійснюється покроково. Для того, щоб проаналізувати новий відеофайл користувач обирає його у файловому менеджері та передає його на вхід обчислюваного конвеєра. Переданий файл оброблюється підсистемою препроцесингу відео, з нього зчитуються кадри та конвертуються в модуль HSV [50]. Далі обчислюється середнє значення яскравості для кадру. Після цього отримані значення аналізуються через перетворення Фур'є [51; 53; 54] і з них виділяється сигнал. Підсистема обробки сигналу обчислює показники варіабельності серцевого

ритму та частоти серцевих скорочень, і передає параметри сигналу на систему візуалізації. Підсистема візуалізації за допомогою бібліотеки `matplotlib` буде кардіоритмограму та відображає користувачеві попередньо розраховані показники діяльності серцево-судинної системи [56]. Ці дані за допомогою відповідної підсистеми заносяться в базу даних. Графічний файл з кардіоритмограмою зберігається на диску окремо, в папці `output_img`, щоб не перевантажувати базу даних.

Додатково програма дає змогу переглядати дані про вже проведені тести та вносити нових осіб в базу.

Оскільки додаток розроблено на Python, то першочергово він запускався і тестувався через консоль. Проте для взаємодії непідготовленого користувача з додатком та роботи з базою даних було прийнято рішення про розробку простого графічного інтерфейсу користувача. Популярні фреймворки графічного інтерфейсу для Python мають одну спільну рису, оскільки вони покладаються на існуючі набори інструментів, написані іншими мовами, крім Python, і надають прив'язки та необхідні перетворення для забезпечення взаємодії з модулями Python. Python включає в себе графічний пакет під назвою Tkinter, який використовує сторонній інструментарій Tk GUI [57]. Фреймворк відносно легкий з точки зору використання ресурсів. Так як не передбачалась розробка складного інтерфейсу користувача, то було вирішено зупинити вибір на вбудованому інструментарії.

При запуску додатку користувач побачить перед собою панель керування, яка містить кнопки для основних операцій (рис. 3.2)

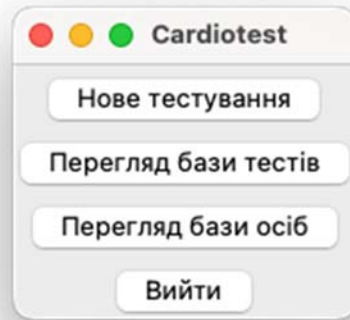


Рис.3.2. Панель керування додатку

При натисканні кнопки «Нове тестування» користувачу буде запропоновано обрати відеофайл, який потрібно проаналізувати. Після вибору файлу він передається на підсистему аналізу. Хід виконання відображається в консолі (рис.3.3).



Рис. 3.3. Хід виконання аналізу відеофайлу

По завершенню аналізу користувачу буде запропоновано ввести інформацію про особу, для якої проводиться тестування. Інформація вноситься за допомогою окремого вікна (рис.3.4)

Додати особу

Внесіть дані про нову особу

Прізвище
Пугач

Ім'я
Тетяна

По батькові
Валеріївна

Дата народження
14-06-1988

Стать
ж

Чи мала/має особа захворювання серцево-судинної системи
0

Додаткова інформація

Додати особу в базу Додати

Рис.3.4 Вікно додавання інформації про особу

У даному вікні потрібно внести наступну інформацію про особу

1. Прізвище, ім'я та по батькові
2. Дата народження
3. Стать
4. Чи має особа захворювання серцево-судинної системи?
5. Додаткова інформація

Усі поля є опціональними щодо заповнення.

Після внесення інформації про особу в базу – відображається вікно з результатами тестування (рис.3.5.)

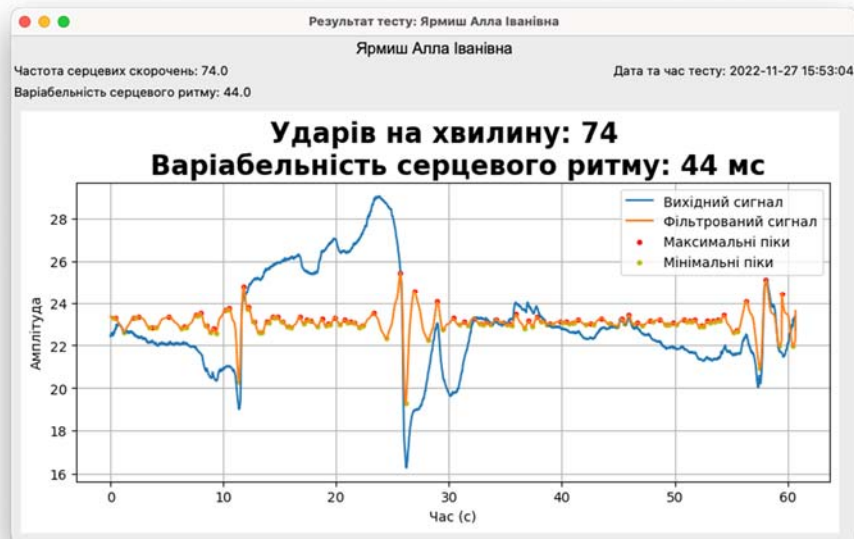


Рис.3.5. Вікно результатів тестування

Вікно містить інформацію про особу, дату тестування, показники роботи серцево-судинної системи та кардіоритмограму.

Результати тестувань зберігаються в базі даних та готові до перегляду у будь-який момент (рис.3.6)

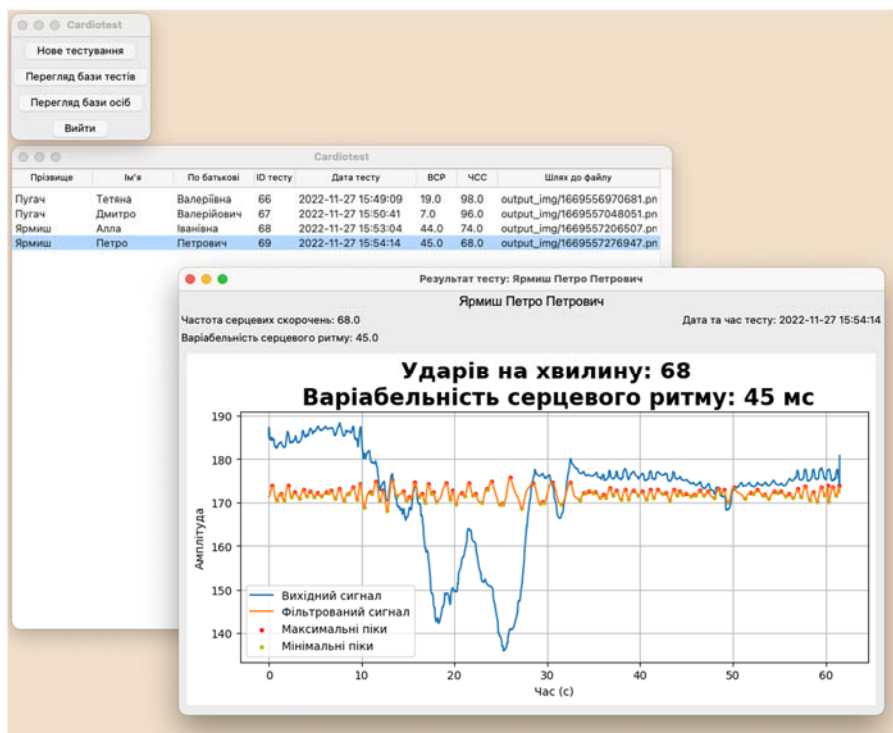


Рис. 3.6. Перегляд попередніх результатів тестування

Додатково програма дає змогу переглядати дані про осіб які проходили тестування.

Persons						
Прізвище	Ім'я	По батькові	Дата народження	Стать	Чи має сі	Додаткова інформація
Пугач	Тетяна	Валеріївна	14-06-1988	ж	0	
Пугач	Дмитро	Валерійович	01-05-1993	ч	0	
Ярмиш	Алла	Іванівна	01-09-1963	ж	1	
Ярмиш	Петро	Петрович	16-06-1966	ч	1	

Рис. 3.7. Вікно перегляду інформації про осіб

Розроблена програма використовує для роботи ряд бібліотек мови Python. Частина із них є складовими стандартної бібліотеки Python, частина є сторонніми і потребує додаткової інсталяції.

Структура модулів програми показана на рис. 3.8.

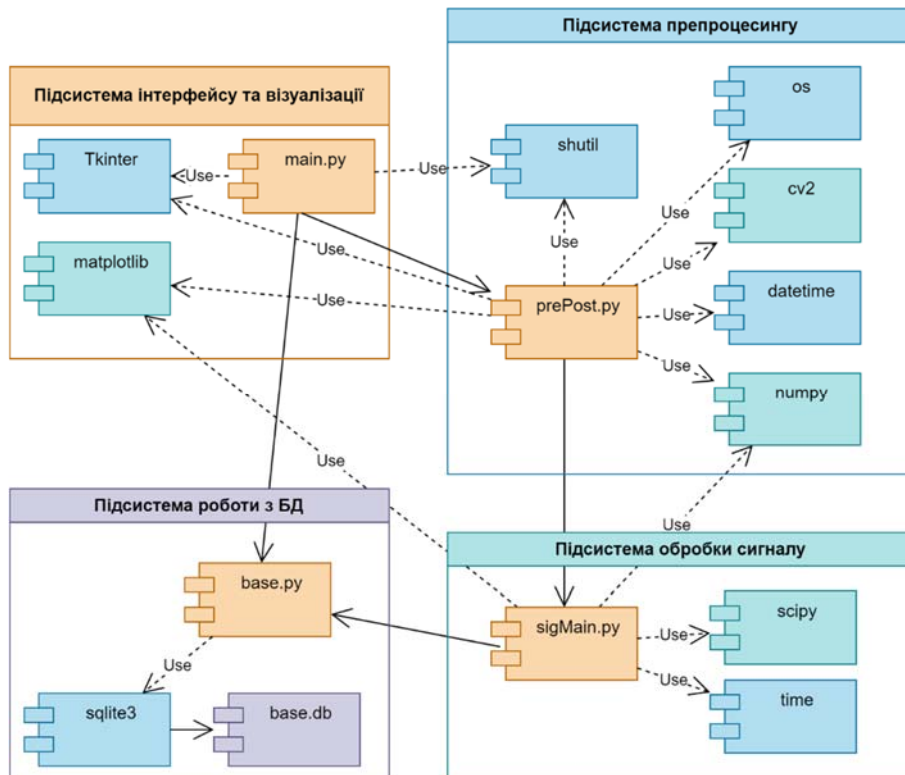


Рис.3.8. Структура модулів програми

До основних модулів відносяться:

1. `main.py` – головний модуль, містить підсистему графічного інтерфейсу користувача та підсистему візуалізації.
2. `prePost.py` – містить підсистему препроцесингу відеофайлів. Отримує дані з головного модуля та передає дані модулю обробки.
3. `sigMain.py` – містить підсистему обробки сигналу, розраховує показники діяльності серцево-судинної системи. Отримує дані з підсистеми препроцесингу, передає їх в підсистеми роботи з БД та візуалізації.
4. `base.py` – містить функції для роботи з базою даних. Взаємодіє з головним модулем та модулем обробки сигналу.

До стандартних бібліотек, яких потребує програма відносяться:

1. `Tkinter` – містить класи для побудови графічного інтерфейсу програми;
2. `time` – бібліотека для роботи з часом;
3. `datetime` – бібліотека для конвертації та роботи з часом і датою;
4. `os` – бібліотека, що містить функції для роботи з деревом каталогів;
5. `shutil` – використовується для маніпуляцій з файлами;
6. `sqlite3` – застосовується для доступу до бази даних.

Всі ці бібліотеки входять до дистрибутиву мови Python та не потребують встановлення.

Бібліотеки `matplotlib`, `numpy`, `scipy` та `opencv` описані в попередніх розділах. Їх застосування передбачає встановлення, оскільки вони не поставляються разом з Python. Як правило встановлення сторонніх бібліотек прийнято виконувати через утиліту командного рядка. На прикладі бібліотеки `numpy` це виглядає наступним чином:

```
pip install numpy
```

Дана команда встановить `numpy` та всі супутні залежності. Аналогічно встановлюються інші бібліотеки.

Для зручнішого розгортання та подальшого перенесення програми разом з потрібними бібліотеками доцільно створити віртуальне середовище за допомогою модуля `venv`.

Модуль `venv` підтримує створення легких «віртуальних середовищ», кожне з власним незалежним набором пакетів Python, встановлених у каталогах сайту. Віртуальне середовище створюється поверх існуючої інсталяції Python, відомої як «базова», і за бажанням може бути ізольоване від пакетів у базовому середовищі.

При використанні у віртуальному середовищі звичайні інструменти інсталяції, такі як `pip`, встановлюють пакети Python у віртуальне середовище.

Створення віртуального середовища для розробленого додатку здійснюється командою

```
python3 -m venv ./cardiotests
```

Виконання цієї команди створює цільовий каталог (створюючи будь-які батьківські каталоги, які ще не існують) і розміщує в ньому файл `pyvenv.cfg` із домашнім ключем, що вказує на інсталяцію Python, з якої було запущено команду (загальна назва цільового каталогу є `.venv`). Він також створює підкаталог `bin` (або сценаріїв у Windows), що містить копію/символьне посилання двійкового/двійкових файлів Python (відповідно до платформи чи аргументів, що використовуються під час створення середовища). Він також створює (спочатку порожній) підкаталог `lib/pythonX.Y/site-packages` (у Windows це `Lib\site-packages`). Якщо вказано існуючий каталог, він буде використаний повторно. Після створення віртуального середовища потрібно встановити перелічені вище сторонні бібліотеки.

Таким чином, повністю для розгортання програми необхідно встановлення мови Python (версії не нижче 3) та набору пакетів. Встановлення цих пакетів забезпечується виконанням команди

```
pip install opencv-python matplotlib numpy scipy
```

Запуск програми здійснюється типово для додатків написаних на Python командою

```
python main.py
```

Після цього відобразиться головне вікно програми (рис.3.2) і можна буде обрати потрібний відеофайл для аналізу.

3.2. Підсистема обробки та зберігання даних

Розглянемо особливості реалізації підсистеми зберігання та обробки даних. Особливостями предметної області є потреба зберігання інформації про осіб та вимірювання, які вони проводять. Можна виділити наступні сутності предметної області [58]:

1. Особи (персони) – сутність для зберігання даних про осіб, має атрибути: ідентифікатор, прізвище, ім'я, по батькові, дата народження, стать, чи є серцеві захворювання, додаткова інформація.

2. Тестування (вимірювання) – сутність для зберігання даних про конкретні виміри показників серцево-судинної системи через камері, має атрибути: ідентифікатор, часова мітка, варіабельність серцевого ритму, частота серцевих скорочень, пульс, графік та вихідний відеофайл.

Графічно дану концептуальну модель можна представити у вигляді діаграми «сутність-зв'язок».

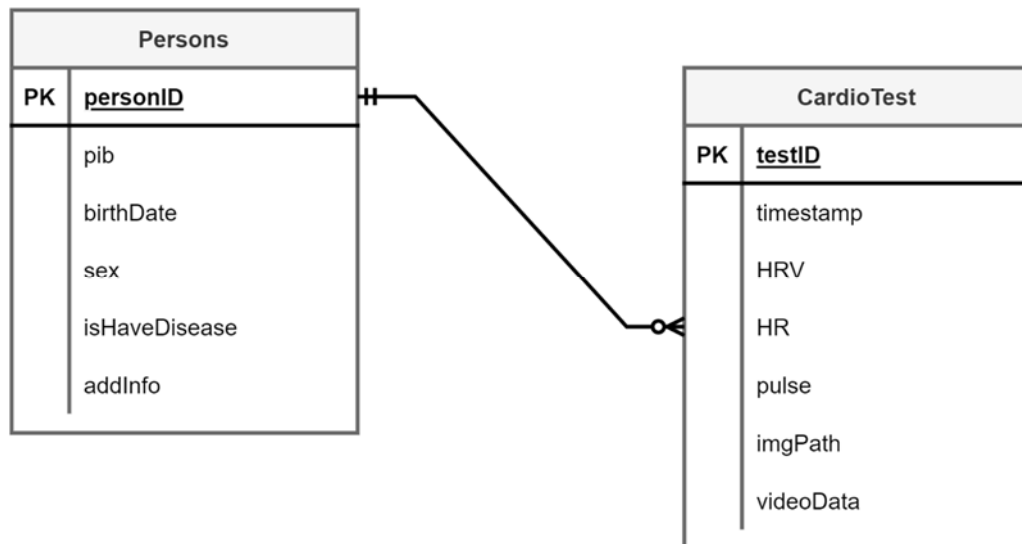


Рис. 3.8. Діаграма «сутність-зв'язок» предметної області

Оскільки в предметній області одному екземпляру сутності «Особа» може відповідати декілька екземплярів сутності «Тестування», то дані сутності пов'язані зв'язком один-до-багатьох [58]. Оскільки для існування сутності «Особа» існування сутності «Тестування» не є абсолютно необхідним, то сутність «Тестування» є необов'язковою по відношенню до сутності «Особа», що відображено на діаграмі.

Виходячи з вимог бізнес логіки було обрано для реалізації бази даних систему керування базами даних SQLite [59]. Дана СКБД має ряд переваг та особливостей, описаних нижче.

SQLite — це внутрішня бібліотека, яка реалізує автономний, безсерверний, нульової конфігурації, транзакційний механізм баз даних SQL. Код для SQLite знаходиться у відкритому доступі, таким чином, безкоштовний для використання для будь-яких цілей, комерційних або приватних. SQLite є найбільш широко розгорнутою базою даних у світі [59].

На відміну від більшості інших баз даних SQL, SQLite не має окремого серверного процесу. SQLite читає і записує безпосередньо на звичайний диск. Повна база даних SQL з декількома таблицями, індексами, тригерами та видами містяться в одному файлі диска. Формат файлу бази даних кросплатформний — можна вільно копіювати базу даних між 32-розрядними і 64-розрядними системами або між архітектурами big-endian і little-endian.

При включених всіх функціях розмір бібліотеки може бути менше 750КБ, в залежності від цільової платформи і налаштувань оптимізації компілятора. Існує компроміс між використанням пам'яті та швидкістю. SQLite зазвичай працює швидше, чим більше пам'яті ви даєте йому. Проте продуктивність зазвичай непогана навіть в середовищах з малою пам'яттю.

SQLite дуже ретельно тестується перед кожним випуск і має репутацію дуже надійного. Велика частина вихідного коду SQLite присвячена виключно тестуванню і верифікації

Оскільки база даних SQLite не вимагає адміністрування, то він добре працює в пристроях, які повинні працювати без експертної підтримки людини. SQLite добре підходить для використання в мобільних телефонах, приставках, телевізорах, фотоапаратах, годинниках, дронах, медичних приладах і роботах.

SQLite часто використовується як формат файлів на диску для настільних додатків, таких як системи контролю версій, інструменти фінансового аналізу, набори каталогізації та редагування медіа, САПР пакети, програми для ведення записів тощо. Цей підхід має багато переваг, включаючи покращену продуктивність, зниження вартості та складності, і підвищену надійність.

Багато додатків використовують SQLite як кеш відповідного вмісту з корпоративною РСКБД. Це зменшує затримку, оскільки більшість запитів тепер виникають до локального кешу та уникають. Це також знижує навантаження в мережі і на центральному сервері баз даних. І в багатьох випадках, це означає, що додаток на стороні клієнта може продовжувати працювати протягом перебоїв в роботі мережі.

База даних SQLite полегшує передачу даних між системами навіть коли кінцеві точки мають різний розмір слів та/або порядок байтів. Дані можуть являти собою складне поєднання великих двійкових блоків, тексту та малих числових або логічних значень.

SQLite не потрібно «встановлювати» перед тим, як він буде використовуватися. Процедури «налаштування» немає. Немає серверного

процес, який потрібно запустити, зупинити або налаштувати. Не потрібно, щоб адміністратор створював новий екземпляр бази даних або призначав його дозволи на доступ до користувачів. SQLite не використовує файли конфігурації.

Більшість механізмів баз даних SQL реалізовані у вигляді окремого сервера процес. Програми, які хочуть отримати доступ до бази даних, обмінюються даними з сервером за допомогою міжпроцесного зв'язку (як правило, TCP/IP) для надсилання запитів на сервер та отримання зворотних результатів. SQLite таким чином не працює. За допомогою SQLite процес, який хоче отримати доступ до бази даних, читає і записує безпосередньо з файлів бази даних на диску. Основною перевагою є те, що немає окремого серверного процесу для інсталяції, настроювання, ініціалізації, керування та усунення несправностей. Програми, які використовують SQLite, не потребують адміністративної підтримки для налаштування обробника баз даних перед їх запуском. Будь-яка програма який може отримати доступ до диска, може використовувати базу даних SQLite.

База даних SQLite - це єдиний звичайний файл диска, який можна знайти будь-де в ієрархії каталогів. Якщо SQLite вміє читати файл диска тоді він може читати що завгодно в базі даних. Якщо файл та його каталог можна переписати, тоді SQLite може змінити що завгодно в базі даних.

Інші рушії баз даних SQL, як правило, зберігають дані у вигляді великої колекції файлів. Часто ці файли знаходяться в стандартному місці, до якого тільки сам движок баз даних може отримати доступ. Це робить дані більш безпечними, але також ускладнює доступ до них. Деякі рушії баз даних SQL надають можливість запису безпосередньо на диск і обходу файлової системи. Це забезпечує додаткову продуктивність, але за рахунок значної складності налаштування та обслуговування. Оптимізована під розмір, вся бібліотека SQLite має розмір менше 500 КБ. Більшість інших движків баз даних SQL набагато більші за цей.

Більшість рушіїв баз даних SQL використовують статичну типізацію. Пов'язаний тип даних з кожним стовпцем у таблиці та лише значеннями цього конкретного типу даних дозволяється зберігати в цьому стовпці. SQLite послаблює це обмеження за допомогою маніфестної типізації. У маніфестній типізації тип даних є властивістю самого значення, а не стовпця, у якому зберігається значення. SQLite, таким чином, дозволяє користувачеві зберігати будь-яке значення будь-якого типу даних у будь-якому стовпці незалежно від оголошеного типу цієї колонки. Більшість інших рушіїв баз даних SQL виділяють фіксований обсяг дискового простору для кожного рядка в більшості таблиць. SQLite, навпаки, використовує лише обсяг дискового простору, насправді потрібний для зберігання інформації. Використання SQLite записів змінної довжини має ряд переваг. Очевидно, що це призводить до менших файлів бази даних. Це також робить запуск бази даних швидшим, оскільки менше інформації для переміщення до та з диску.

Схема розробленої бази даних наведена на рис.3.8.

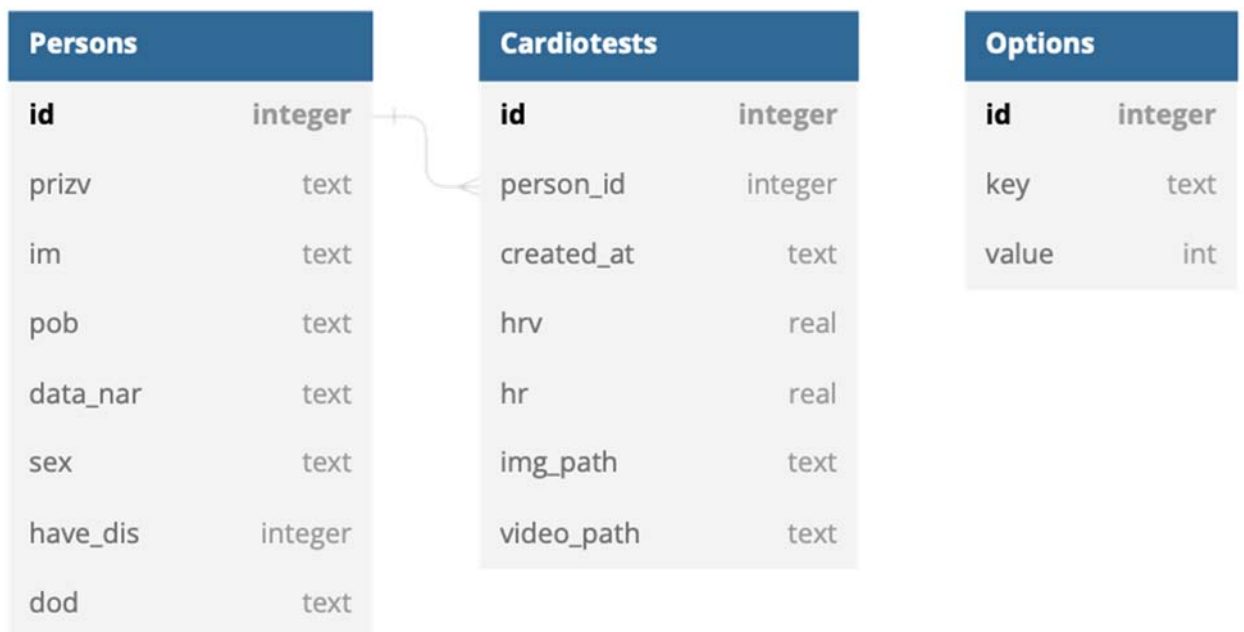


Рис. 3.8. Схема розробленої бази даних

Вона містить три таблиці Persons, Cardiotests та Options. Враховуючи специфіку типів даних SQLite, опис та призначення полів таблиць наведено в табл.3.1.

Таблиця 3.1

Структура та призначення полів таблиці Persons

Назва поля	Тип даних	Опис, призначення
id	integer	Ідентифікатор особи, автоінкремент, первинний ключ
prizv	text	Прізвище особи
im	text	Ім'я особи
rob	text	По батькові особи
data_nar	text	Дата народження
sex	text	Стать
have_dis	integer	Чи має особа серцеві захворювання, {0,1}
dod	text	Додаткова інформація

Таблиця 3.2

Структура та призначення полів таблиці Cardiotests

Назва поля	Тип даних	Опис, призначення
id	integer	Ідентифікатор тесту, автоінкремент, первинний ключ
person_id	integer	Ідентифікатор особи, зовнішній ключ
created_at	text	Часова мітка
hrv	real	Обчислена варіабельність серцевого ритму
hr	real	Обчислена частота серцевих скорочень
image_path	text	Шлях до графічного файлу з результатами
video_path	text	Шлях до вихідного відеофайлу

Дані таблиці пов'язані відношенням один-до-багатьох через наступне обмеження зовнішнього ключа:

Persons.id= Cardiotests.person_id

Структура та призначення полів таблиці Options

Назва поля	Тип даних	Опис, призначення
id	integer	Ідентифікатор опції, автоінкремент, первинний ключ
key	text	Ключ, текстовий ідентифікатор опції
value	integer	Значення опції

Дана таблиця призначена для зберігання параметрів, які потрібні для конфігурування додатку, а також використовується як сховище для тимчасових змінних. Важливою особливістю sqlite у цьому контексті є маніфестна декларація типів даних, оскільки хоча більшість значень в стовпці value і є цілими числами, деякі з них представляють собою рядки, які було б неможливо зберігати в більшості систем керування базами даних з такою схемою БД.

Для реалізації універсального механізму доступу до параметрів в БД написані функції `set_option(key, value)` та `get_option(key)`, які відповідно дозволяють встановлювати значення за певним ключем, або зчитувати його. Дані функції використовуються іншими функціями, що працюють з Options як інтерфейс. Їх лістинг наведено нижче:

```
def set_option(key, value):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
UPDATE options SET value = ? WHERE key = ?
''', (value, key))
    conn.commit()
    conn.close()

def get_option(key):
    conn = create_connection()
    sql = 'SELECT value FROM options WHERE key = ?'
    cur = conn.cursor()
    cur.execute(sql, (key,))
    rows = cur.fetchall()
    conn.close()
    return rows[0][0]
```

Для встановлення зв'язку з БД використовується додаткова функція `create_connection()`, яка повертає об'єкт типу `connection`.

3.3. Статистична оцінка показників, одержаних роботи в результаті роботи програми

Для оцінки коректності вимірювань можна порівняти результати, отримані за допомогою програми з результатами, отриманими за допомогою смарт-пристроїв, які мають функцію визначення частоти серцевих скорочень (вимірювання пульсу) з використанням методу фотоплетизмографії [60]. Було проведено 50 записів з камери смартфона з одночасним вимірюванням пульсу за допомогою Apple Watch 6 та Xiaomi Amazfit Verge. Тестова група включала осіб різного віку, статі та з різним рівнем здоров'я серцево-судинної системи. Заміри проводились протягом однієї хвилини, після години перебування в спокої, і в одному й тому ж приміщенні.

Таблиця 3.4

Результати вимірювань пульсу, які одержані з камери та смарт-годинників

Камера	84	82	78	87	102	77	103	66	71	85	85	84	82
Apple Watch 6	82	83	77	89	104	77	102	67	69	85	86	83	80
Xiaomi Amazfit Verge	83	82	79	91	103	75	101	68	67	86	86	82	81
Камера	88	78	80	90	100	76	68	72	84	87	81	70	86
Apple Watch 6	87	80	80	88	102	75	68	73	85	86	81	68	86
Xiaomi Amazfit Verge	88	80	78	86	103	76	69	71	84	84	83	68	84
Камера	97	92	105	74	101	91	88	87	83	68	106	80	
Apple Watch 6	96	91	104	74	102	91	87	89	85	70	105	82	
Xiaomi Amazfit Verge	96	90	105	76	103	89	89	88	84	68	107	81	
Камера	100	79	88	87	77	84	96	82	99	76	101	78	
Apple Watch 6	102	81	89	85	78	82	94	83	101	74	99	78	
Xiaomi Amazfit Verge	104	80	90	85	77	82	94	83	103	72	100	77	

Графічно результати вимірів показано на рис. 3.9. З метою покращення візуального сприйняття показано тільки перші 15 результатів. Як видно з

графіку, показники отримані з камери смартфона та отримані за допомогою смарт годинників відрізняються несуттєво.

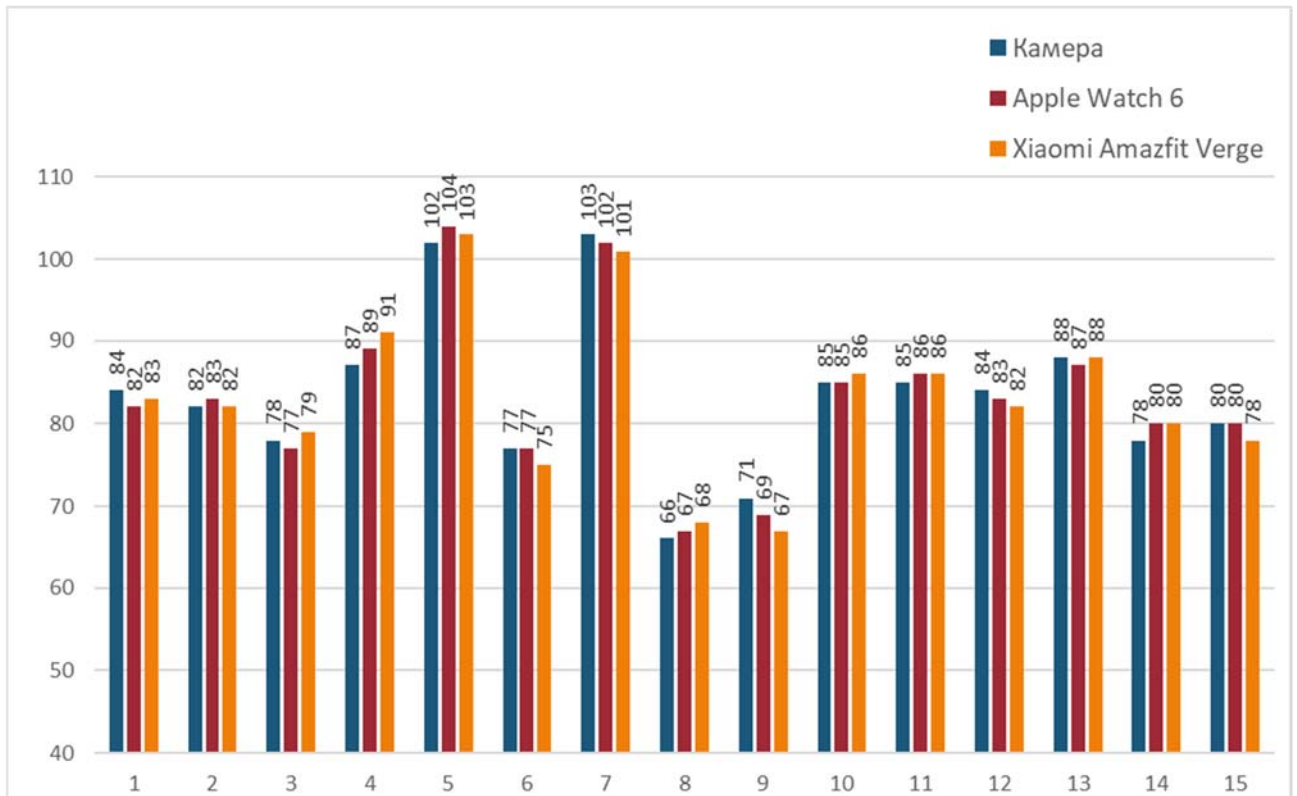


Рис. 3.9. Графічні результати вимірювань пульсу, які одержані з камери та смарт-годинників

Для статистичного порівняння одержаних груп величин обчислимо для кожної групи середнє значення, медіану, 0,25 та 0,75 квантилі, дисперсію та середньоквадратичне відхилення [61; 62].

r -перцентиль – це показник, який відповідає значенню, нижче якого знаходяться r % спостережуваної змінної, впорядкованої за зростанням значення. У статистиці як правило розглядаються 25 та 75 перцентилі (1-й та 3-й квантилі), а також 50 перцентиль – медіана [63].

Дисперсія S^2 дорівнює середньому значенню квадрата відхилення величини x_i від її середнього значення \bar{x} [61; 62]. У даному випадку ми маємо вибірку, тому для кожної групи буде обчислюватись незміщена оцінка дисперсії за формулою (3.1)

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (3.1)$$

Середньоквадратичне відхилення обчислюється як квадратний корінь з дисперсії і є мірою розкиду деякої величини від її середнього значення [61; 62].

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.2)$$

Описові статистики для даних наведено в табл.3.6.

Таблиця 3.6

Описові статистики для одержаних даних

Показник	Камера	Apple Watch 6	Xiaomi Amazfit Verge
Середнє значення	85,3	85,3	85,22
0,25 квантиль	78	78	77,75
Медіана	84	85	84
0,75 квантиль	91,25	91	90,25
Дисперсія	107,1531	108,3367	117,8078
Стандартне відхилення	10,35148	10,40849	10,85393

Із таблиці видно, що значення всіх статистичних показників дуже близькі. Покажемо це у вигляді діаграми розсіювання [64], яка наочно демонструє їх подібність.

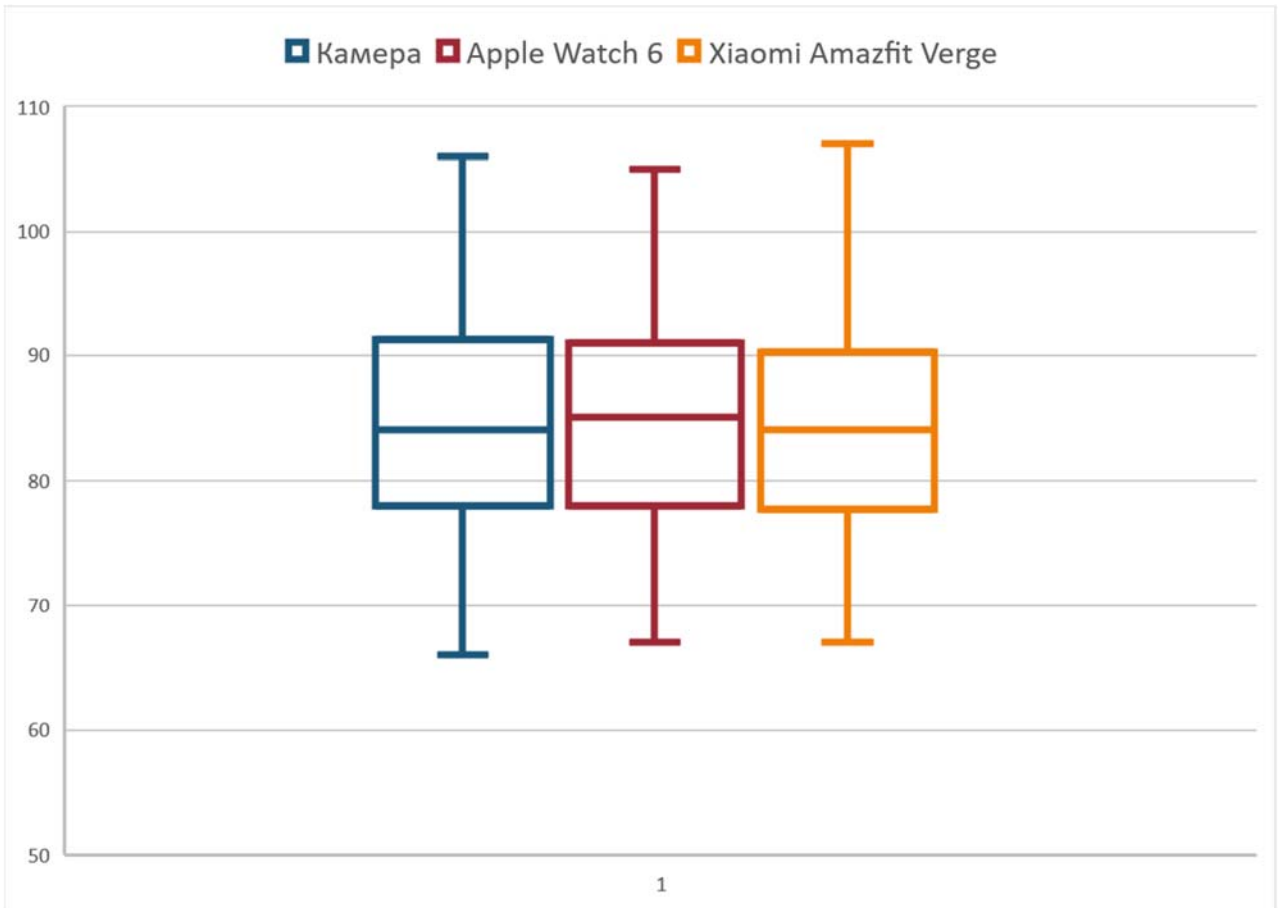


Рис. 3.10. Порівняння результатів вимірювань, отриманих із різних пристроїв

На рис. 3.11. наведено графіки середнього абсолютного відхилення значень частоти серцевих скорочень, отриманих за допомогою камери смартфона, від показників смарт годинників, а також відносна похибка.

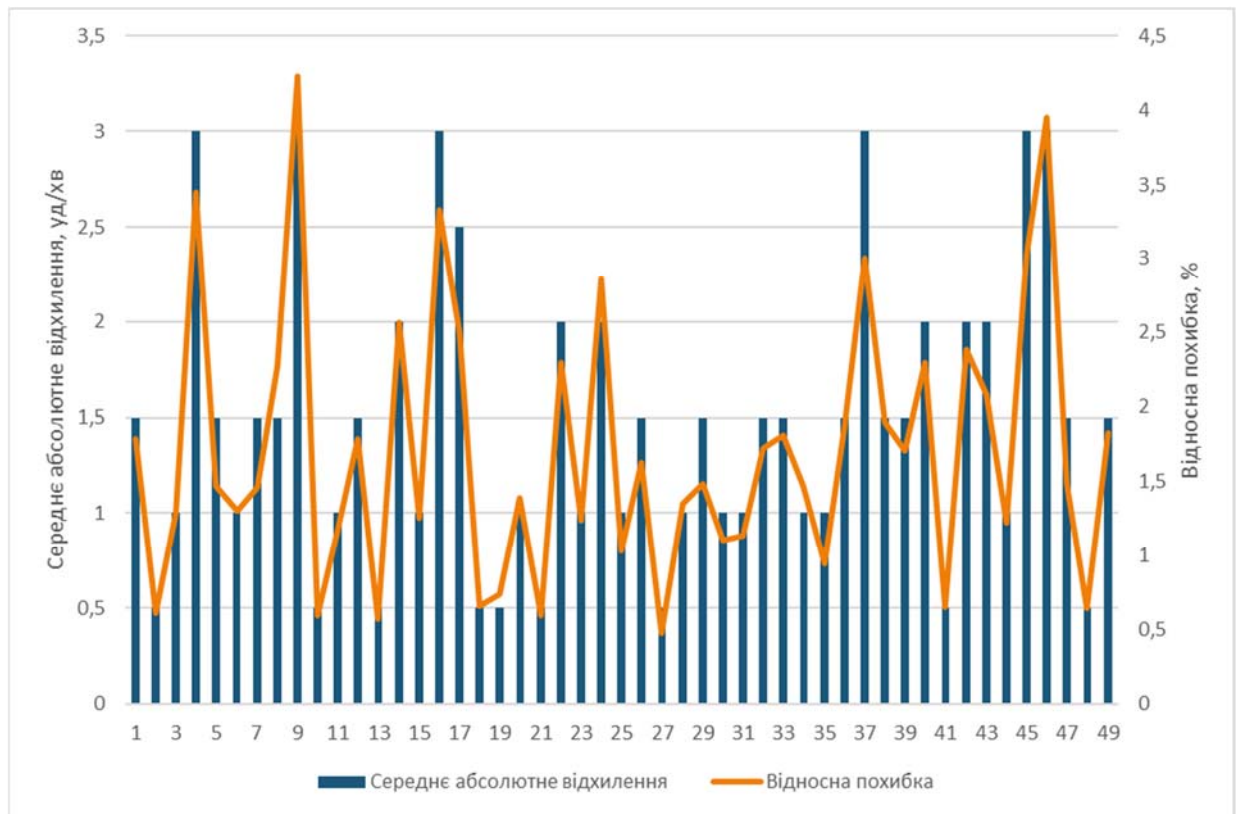


Рис. 3.11 Середнє абсолютне відхилення та відносна похибка отриманих результатів

Середнє абсолютне відхилення d_i обчислене за формулою [61; 62] (3.3)

$$d_i = \left| x_{1i} - \frac{1}{2}(x_{2i} + x_{3i}) \right| \quad (3.3)$$

де

x_{1i} – значення величини пульсу, одержані за допомогою камери, x_{2i} , x_{3i} – відповідно значення, одержані з Apple Watch та Xiaomi Amazfit.

Відносна похибка δ визначена як

$$\delta_i = \frac{d_i}{x_{1i}} \cdot 100\% \quad (3.4)$$

Як бачимо з графіків, середнє абсолютне відхилення частоти серцебиття та відносна похибка знаходяться в межах 3-4% відсотків для тестової групи. В цілому можна сказати, що показники, які отримані за допомогою камери смартфона, є достатньо близькими до показників, які отримані за допомогою смарт годинників. Звертаємо увагу, що з медичної точки зору результати не

можуть бути використані для постановки діагнозу, і для точного діагнозу необхідно звертатись до лікаря та проводити клінічне обстеження.

Висновки до розділу 3

1. Розроблено програмну систему для аналізу відеоданих з метою визначення показників роботи серцево-судинної системи.
2. Охарактеризовано процес роботи з програмною системою, побудовано діаграму послідовності.
3. Спроектовано підсистему зберігання даних з урахуванням вимог предметної області, та виконано вибір реляційної системи керування базами даних.
4. Описано архітектуру програмної системи, її підсистем та модулів. Наведений порядок розгортання системи разом із супутніми залежностями.
5. Здійснено статистичну оцінку та порівняння результатів роботи розробленого програмного забезпечення з пристроями, функціонування яких також ґрунтується на застосуванні метод фотоплетизмографії.

ВИСНОВКИ

В ході виконання випускної кваліфікаційної роботи було здійснено вибір математичного апарату, на основі якого розроблено архітектуру та реалізацію програмної системи аналізу відеоданих для визначення показників діяльності серцево-судинної системи.

У процесі роботи над дослідженням було вирішено наступні завдання:

1. Виконано огляд методів визначення показників роботи серцево-судинної системи, таких як частота серцевих скорочень, серцевий індекс, хвилинний об'єм кровообігу та варіабельність серцевого ритму.

2. Розроблено концептуальну модель програмного забезпечення для аналізу відеоданих. Структурно розроблена модель включає підсистеми препроцесингу відео, обробки сигналу, візуалізації та роботи з базою даних.

3. Обрано математичний апарат для виділення та аналізу сигналу, одержаного шляхом перетворення відеоданих. У якості математичного та алгоритмічного забезпечення було обрано швидке перетворення Фур, яке дозволяє ефективно вирішувати поставлені завдання.

4. Здійснено огляд та побудову програмного стеку технологій для реалізації системи. Обрано технології на базі мови високого рівня Python. Зокрема для роботи з відеопотоком обрано бібліотеку OpenCV, для математичного забезпечення – бібліотеку numpy.

5. Спроектовано модель зберігання даних для забезпечення функціонування програмної системи. Відповідно до вимог предметної області реалізовано реляційну базу даних на основі технології sqlite.

6. На основі вибраного стеку технологій розробки реалізовано програмне забезпечення для аналізу відеоданих, одержаних за допомогою камери смартфона, для визначення показників діяльності серцево-судинної системи. Програмне забезпечення відповідає функціональним вимогам та є кросплатформним.

7. Виконано порівняльний аналіз ефективності визначення показників діяльності серцево судинної системи за допомогою камери смартфона з даними, одержаними за допомогою альтернативних засобів

вимірювання, зокрема смарт-годинників. Проведений аналіз показав подібність результатів одержаних за допомогою програми, та з пристроїв, оснащених відповідними апаратними датчиками та сенсорами.

В цілому, можна зробити висновок, що показники, які отримані за допомогою камери смартфона, є достатньо близькими до показників, які отримані за допомогою інших пристроїв. Це свідчить про достатню ефективність та адекватність алгоритму. Отримані результати можуть бути застосовані для повсякденного спостереження за станом здоров'я з профілактичною, персоналізованою та прогностичною метою. Проте, слід зазначити, що з медичної точки зору результати не можуть бути використані для постановки діагнозу, і для точного діагнозу необхідно звертатись до лікаря та проводити клінічне обстеження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Global Burden of Cardiovascular Diseases and Risk Factors, 1990–2019. Institute for Health Metrics and Evaluation. URL: <https://www.healthdata.org/research-article/global-burden-cardiovascular-diseases-and-risk-factors-1990–2019>.
2. Ukraine. Institute for Health Metrics and Evaluation. URL: <https://www.healthdata.org/ukraine>
3. Державна служба статистики України URL: https://ukrstat.gov.ua/operativ/operativ2021/ds/krops/arh_krops2021_u.html.
4. Рівень смертності в Україні: від чого найчастіше помирають громадяни. Слово і Діло. URL: <https://www.slovoidilo.ua/2021/11/23/infografika/suspilstvo/riven-smertnosti-ukrayini-choho-najchastishe-pomyrayut-hromadyanu>
5. Terenda N. A. Основні тенденції та прогностні оцінки загальної та первинної захворюваності на ішемічну хворобу серця в Україні. Вісник соціальної гігієни та організації охорони здоров'я України. 2016. № 3. URL: <https://doi.org/10.11603/1681-2786.2016.3.7007>.
6. Профілактика серцево-судинних захворювань • Чернігівська міська лікарня № 3. Чернігівська міська лікарня № 3. URL: <https://ml3.lic.org.ua/zdorovia-vid-a-do-ia/profilaktyka-sertsevo-sudynnykh-zakhvoriuvan/>
7. Крючко І., Петросян Л., Стенцель Й. Дослідження функціонального стану серцево-судинної системи організму людини. Вісник Східноукраїнського національного університету імені Володимира Даля. 2016. С. 97–102.
8. Шинкарук О., Тайболіна Л. Функціональний стан серцево-судинної системи веслувальників високої кваліфікації в процесі інтенсивної змагальної діяльності. Спортивна медицина, фізична терапія та ерготерапія. 2020. № 1. С. 49–60. URL: <https://doi.org/10.32652/spmed.2020.1.49-60>

9. Профілактика серцево-судинних захворювань • Чернігівська міська лікарня № 3. Чернігівська міська лікарня № 3. URL: <https://ml3.lic.org.ua/zdorovia-vid-a-do-ia/profilaktyka-sertsevo-sudynnykh-zakhvoriuvan/>
10. Як визначити оптимальну вагу: формула індексу маси тіла. Міністерство охорони здоров'я України. URL: <https://moz.gov.ua/article/health/jak-viznachtsi-optimalnu-vagu-formula-indeksu-masi-tila>.
11. Індекс маси тіла: як зрозуміти, що вага у нормі. znaimo: Платформа про здорове шкільне харчування. URL: https://znaimo.gov.ua/indeks_masy_tila_yak_zrozumity_shcho_vaha_u_normi
12. Глобальні рекомендації щодо фізичної активності для здоров'я. Всесвітня організація охорони здоров'я. URL: <https://apps.who.int/iris/bitstream/handle/10665/44399/9789241599979-ukr.pdf?sequence=25&isAllowed=y>
13. Маліков М., Сват'єв А., Богдановська Н. Функціональна діагностика у фізичному вихованні і спорті : навч. посіб. Запоріжжя : ЗДУ, 2006. 227 с.
14. Дослідження пульсу й артеріального тиску : навч. посіб. / упоряд.: Т. Ащеулова, Т. Амбросова, В. Смирнова. Харків : ХНМУ, 2018. 20 с.
15. Візір В., Приходько І., Деміденко О. Електрокардіографія. Функціональні ЕКГ проби. Добове моніторування артеріального тиску. Холтерівське моніторування ЕКГ. Аналіз варіабельності серцевого ритму (BCP). Функціональна діагностика в пульмонології : навч. посіб. Запоріжжя, 2014. 116 с.
16. Серцевий ритм – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Серцевий_ритм
17. Hurst's The Heart: Manual of Cardiology / R. A. O'Rourke et al. McGraw-Hill Professional, 2001. 798 p.

18. Benson R. T., Connolly D. Heart Rate Training. Human Kinetics, 2019. 272 p.
19. Atwal S., Porter J., MacDonald P. Cardiovascular effects of strenuous exercise in adult recreational hockey: the Hockey Heart Study. 3rd ed. 2002. 503 p.
20. Target Heart Rate and Estimated Maximum Heart Rate | Physical Activity | CDC. Centers for Disease Control and Prevention. URL: <https://www.cdc.gov/physicalactivity/basics/measuring/hearttrate.htm>
21. Cardiac Stress Test (Exercise Stress Test) - Capital Cardiology Associates. Capital Cardiology Associates. URL: <https://capitalcardiology.com/patient-education/heart-procedures/exercise-stress-test>
22. Mouse Heart Rate in a Human: Diagnostic Mystery of an Extreme Tachyarrhythmia / L. Chhabra et al. Indian Pacing and Electrophysiology Journal. 2012. Vol. 12, no. 1. P. 32–35. URL: [https://doi.org/10.1016/s0972-6292\(16\)30463-6](https://doi.org/10.1016/s0972-6292(16)30463-6)
23. Wohlfart B., Farazdaghi G. R. Reference values for the physical work capacity on a bicycle ergometer for men - a comparison with a previous study on women. Clinical Physiology and Functional Imaging. 2003. Vol. 23, no. 3. P. 166–170. URL: <https://doi.org/10.1046/j.1475-097x.2003.00491.x>
24. Farazdaghi G. R., Wohlfart B. Reference values for the physical work capacity on a bicycle ergometer for women between 20 and 80 years of age. Clinical Physiology. 2001. Vol. 21, no. 6. P. 682–687. URL: <https://doi.org/10.1046/j.1365-2281.2001.00373.x>
25. Сичов О., Жарінов О. Варіабельність серцевого ритму: фізіологічні механізми, методи дослідження, клінічне і прогностичне значення. Компендіум. Методи функціональної діагностики в кардіології. URL: <https://compendium.com.ua/uk/clinical-guidelines-uk/cardiology-uk/section-5-uk/>.

26. Лісун Ю. Б., Углев Є. І. Heart rate variability, applying and methods of analysis. *Pain, Anaesthesia and Intensive Care*. 2020. No. 4(93). P. 83–89. URL: [https://doi.org/10.25284/2519-2078.4\(93\).2020.220693](https://doi.org/10.25284/2519-2078.4(93).2020.220693)
27. Biophysical Characterization of the Underappreciated and Important Relationship Between Heart Rate Variability and Heart Rate / O. Monfredi et al. *Hypertension*. 2014. Vol. 64, no. 6. P. 1334–1343. URL: <https://doi.org/10.1161/hypertensionaha.114.03782>
28. Heart arrhythmia - Symptoms and causes. Mayo Clinic. URL: <https://www.mayoclinic.org/diseases-conditions/heart-arrhythmia/symptoms-causes/syc-20350668>
29. Heart rate as a prognostic risk factor in patients with coronary artery disease and left-ventricular systolic dysfunction (BEAUTIFUL): a subgroup analysis of a randomised controlled trial / K. Fox et al. *The Lancet*. 2008. Vol. 372, no. 9641. P. 817–821. URL: [https://doi.org/10.1016/s0140-6736\(08\)61171-x](https://doi.org/10.1016/s0140-6736(08)61171-x)
30. Серцевий викид – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Серцевий_викид
31. Маліков М. В. Фізіологія фізичних вправ : Навч. посіб. для студ. фак-тів фіз. вихован. вищ. навч. закл. Запоріжжя : ЗДУ, 2003. 112 с.
32. Calculators: Площа поверхні тіла (метод Дюбуа) - MSD Manual Professional Edition. MSD Manual Professional Edition. URL: <https://www.msmanuals.com/uk-ua/professional/multimedia/clinical-calculator/площа-поверхні-тіла-метод-дюбуа>
33. Sörnmo L., Laguna P. Bioelectrical signal processing in cardiac and neurological applications. Academic Press, 2005. 688 p.
34. Heart Rate Variability (HRV) Signal Analysis / ed. by M. V. Kamath, M. Watanabe, A. Upton. CRC Press, 2016. URL: <https://doi.org/10.1201/b12756>
35. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use / M. Malik et al. *European Heart Journal*. 1996. Vol. 17, no. 3. P. 354–381. URL: <https://doi.org/10.1093/oxfordjournals.eurheartj.a014868>

36. Fedotov A. A., Akulov S. A., Timchenko E. V. Methods of Mathematical Analysis of Heart Rate Variability. Biomedical Engineering. 2020. Vol. 54, no. 3. P. 220–225. URL: <https://doi.org/10.1007/s10527-020-10008-4>
37. Heart rate variability: a review / U. Rajendra Acharya et al. Medical & Biological Engineering & Computing. 2006. Vol. 44, no. 12. P. 1031–1051. URL: <https://doi.org/10.1007/s11517-006-0119-0>
38. A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects / G. Lu et al. Journal of Medical Engineering & Technology. 2009. Vol. 33, no. 8. P. 634–641. URL: <https://doi.org/10.3109/03091900903150998>
39. Березанська М. Огляд та сучасний стан датчиків серцевого ритму. Ефективність інженерних рішень у приладобудуванні. С. 26–29.
40. Михайлов В. Вариабельность ритма сердца: опыт практического применения метода. Иваново : Иван. Гос. Мед. Академия, 2002. 290 с.
41. Измерение variability сердечного ритма человека с помощью камеры смартфона. Электронный научный архив УрФУ: Главная страница. URL: <https://elar.urfu.ru/bitstream/10995/43741/1/cai-2016-32.pdf>
42. Діаграми UML для моделювання процесів і архітектури проекту. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
43. Перетворення RGB в HSV | перетворення кольорів. Calculadoras y herramientas en línea - RT. URL: <https://www.rapidtables.org/uk/convert/color/rgb-to-hsv.html>
44. Цифрова обробка аудіо- та відеоінформації у мультимедійних системах : навч. посіб. / О. Дробик та ін. Київ : Наук. думка, 2008. 144 с.
45. Рибальченко М., Єгоров О., Зворикін В. Цифрова обробка сигналів : навч. посіб. Дніпро : НМетАУ, 2018. 79 с.
46. Дасюк Я., Ільків В., Каленюк П. Функції комплексної змінної. Перетворення Фур'є та Лапласа : навч. посіб. Львів : Ін-т змісту і методів навчання, 1999. 271 с.

47. Cooley J. W., Tukey J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*. 1965. Vol. 19, no. 90. P. 297.
URL: <https://doi.org/10.1090/s0025-5718-1965-0178586-1>
48. Fast Fourier Transform (FFT) Background. *Geospatial Solutions & Analytics | Remote Sensing Technology*.
URL: <https://www.l3harrisgeospatial.com/docs/backgroundfastfouriertransform.html>
49. Home - OpenCV. OpenCV. URL: <https://opencv.org>
50. OpenCV: Color conversions. OpenCV documentation index.
URL: https://docs.opencv.org/4.x/de/d25/imgproc_color_conversions.html
51. OpenCV: Fourier Transform. OpenCV documentation index.
URL: https://docs.opencv.org/3.4/de/dbc/tutorial_py_fourier_transform.html
52. NumPy Documentation. NumPy. URL: <https://numpy.org/doc/>
53. numpy.fft.fft – NumPy v1.23 Manual. NumPy.
URL: <https://numpy.org/doc/stable/reference/generated/numpy.fft.fft.html>
54. numpy.fft.ifft – NumPy v1.23 Manual. NumPy.
URL: <https://numpy.org/doc/stable/reference/generated/numpy.fft.ifft.html>
55. Welcome to python.org. Python.org. URL: <https://www.python.org/doc/>.
56. Users guide – Matplotlib 3.6.2 documentation. Matplotlib – Visualization with Python. URL: <https://matplotlib.org/stable/users/index.html>
57. Tkinter Python interface to Tcl/Tk – Python 3.11.0 documentation. URL: <https://docs.python.org/3/library/tkinter.html>.
58. Entity–relationship model - Wikipedia. Wikipedia, the free encyclopedia.
URL: https://en.wikipedia.org/wiki/Entity–relationship_model
59. SQLite Download Page. SQLite Home Page.
URL: <https://sqlite.org/download.html>
60. Сєрова І., Аксьонова І. Організація статистичних спостережень. Харків : Вид-во ХНЕУ, 2008. 274 с.

61. Герич М., Синявська О. Математична статистика. Ужгород : Говерла, 2021. 146 с.

62. Руденко В. М. Математична статистика : навч. посіб. для студ. вищ. навч. закл. Київ : Центр учб. літ., 2012. 304 с.

63. Percentile - Wikipedia. Wikipedia, the free encyclopedia.
URL: <https://en.wikipedia.org/wiki/Percentile>

64. seaborn.boxplot – seaborn 0.12.1 documentation. seaborn: statistical data visualization – seaborn 0.12.1 documentation.
URL: <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

ДОДАТКИ

Додаток А

Копії публікацій

Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво"
Луцьк, 2022. Випуск № 48

141

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-48-22>

УДК 004.94

Ройко Олександр Юрійович¹, к.т.н., викладач

<https://orcid.org/0000-0001-8642-7707>

Пугач Тетяна Валеріївна², студентка

<https://orcid.org/0000-0002-4021-288X>

¹Відокремлений структурний підрозділ «Волинський фаховий коледж Національного університету харчових технологій», м. Луцьк, Україна

²Західноукраїнський національний університет, м. Тернопіль, Україна

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ПОКАЗНИКІВ РОБОТИ СЕРЦЯ З ВИКОРИСТАННЯМ КАМЕРИ СМАРТФОНУ

Ройко О.Ю., Пугач Т.В. Особливості реалізації програмного забезпечення для визначення показників роботи серця з використанням камери смартфона. У роботі описано особливості реалізації програмного забезпечення для визначення варіабельності серцевого ритму та частоти серцевих скорочень. Для формування вихідного сигналу застосовується камера смартфона. Подальша обробка сигналу здійснюється за допомогою швидкого перетворення Фур'є. Отримані дані використовуються для визначення показників роботи серця за допомогою реалізованого програмного забезпечення.

Ключові слова: варіабельність серцевого ритму, частота серцевих скорочень, Python, OpenCV, matplotlib.

Royko O., Puhach T. Peculiarities of implementing the software for determining the performance of the heart using a smartphone camera. The paper describes the features of the implementation of the software for determining the variability of heart rhythm and heart rate. A smartphone camera is used to generate the output signal. Further processing of the signal is carried out using a fast Fourier transform. The obtained data are used to determine the performance of the heart with the help of implemented software.

Keywords: heart rate variability, heart rate, Python, OpenCV, matplotlib.

Постановка наукової проблеми. Захворювання серцево-судинної системи є однією з головних причин людської смертності, як у світі, так і в Україні. Це підтверджують дані ВООЗ [1], а також дослідження провідних науковців [2-3]. Як зазначено у більшості досліджень, для зниження рівня смертності від серцево-судинних захворювань необхідно вести здоровий спосіб життя, уникати стресів, а також проводити своєчасну діагностику ранніх захворювань.

Зараз на ринку існує велика кількість «розумних» пристроїв, які дозволяють пересічному користувачу здійснювати моніторинг показників серцево-судинної системи. До них відносяться смарт годинники та смарт браслети. Звичайно, що точність показників, отриманих за допомогою таких пристроїв, не можна порівнювати з показниками, отриманими за допомогою спеціального медичного обладнання, однак вони дозволяють кожній людині регулярно здійснювати моніторинг власного здоров'я в побутових умовах, що є важливим в профілактиці. Переважно ці пристрої можуть визначати пульс (частоту серцевих скорочень), а також деякі з них можуть визначати варіабельність серцевого ритму [4]. Ці показники при постійному моніторингу можуть давати фахівцю інформацію стосовно здоров'я серцево-судинної системи пацієнта.

На даний час відносно невеликий відсоток населення володіють такими пристроями. Однак смартфон є майже у кожного, тому практичний інтерес становить дослідження можливості розробки програмного забезпечення, яке дозволить отримувати та аналізувати дані про роботу серця зі смартфона.

Ідея полягає в тому, що притискаючи палець до основної камери смартфона є можливість зафіксувати яскравість зображення. В процесі проходження крові через судини в пальцях яскравість зображення змінюється. Така зміна є періодичною і відповідає періодам пульсації кровоносних судин. Тому якщо зафіксувати такі пульсації, ці дані можна обробити та застосувати для визначення показників серцевої діяльності.

Мета дослідження. Необхідно реалізувати програмне забезпечення для отримання деяких показників роботи серцево-судинної системи, таких як, варіабельність серцевого ритму та частота серцевих скорочень. Програмне забезпечення повинно отримувати на вході множину відеофайлів із записом пульсації капілярів у стінках пальців, знятих за допомогою камери смартфона. На виході, після аналізу відеофайлів, необхідно відобразити сигнал у графічному вигляді та показники роботи серця.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Для реалізації поставленого у роботі завдання була розроблена програма на мові Python, з використанням бібліотек OpenCV, matplotlib, numpy та scipy. Її діаграма діяльності показана на рис. 1.

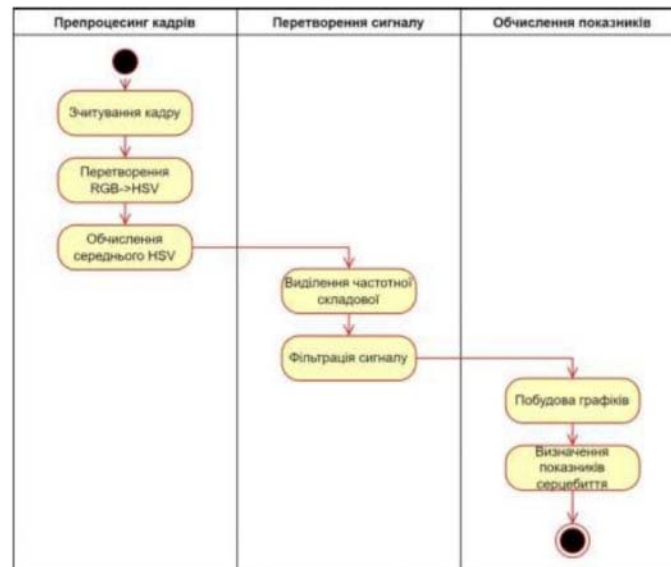


Рис. 1. UML-діаграма діяльності програми

Як видно з діаграми, обчислювальний конвеєр складається з трьох етапів:

1. Препроцесинг кадрів – зчитування та перетворення кадрів для подальшої роботи алгоритму.
2. Перетворення сигналу – виділення із даних, отриманих з кадрів, сигналу.
3. Обчислення показників – визначення показників серцебиття, таких як варіабельність серцевого ритму та частота серцевих скорочень, візуалізація результату.

Детальніше розглянемо кожен з етапів. Для препроцесингу кадрів застосовується бібліотека OpenCV [5]. Дана бібліотека є кросплатформеною, безкоштовною, та містить велику кількість функцій для роботи з відео в реальному часі.

За допомогою OpenCV здійснюється зчитування кадрів відеопотоку та перетворення кожного кадру із колірної моделі RGB в колірну модель HSV. Це необхідно для фіксації періодів зміни яскравості зображення при запису пульсації крові через пальці, що відповідає періодам серцевих скорочень.

Саме ці перепади у яскравості кадрів будемо розглядати як вхідний сигнал. Оскільки модель RGB не несе інформації про яскравість зображення, то кадри потрібно перевести в модель HSV. Для перетворення значення кольору кожного пікселю з колірної моделі RGB в колірну модель HSV використовуються наступні залежності:

$$V = \max(R, G, B)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} \frac{60(G-B)}{V - \min(R, G, B)} & \text{if } V = R \\ \frac{120 + 60(B-R)}{V - \min(R, G, B)} & \text{if } V = G, \\ \frac{240 + 60(R-G)}{V - \min(R, G, B)} & \text{if } V = B \end{cases}$$

де R, G, B – значення інтенсивності кольору (червоного, зеленого та синього) для кожного пікселю, H, S, V – відтінок, насиченість та яскравість відповідно.

На виході отримуємо тривимірні масиви даних із значеннями яскравості, насиченості та відтінку кольору.

Після перетворення кадрів відео в колірну модель HSV відбувається їх усереднення за яскравістю пікселів для кожного кадру.

Наступним кроком є виділення частоти зміни яскравості зображення з вихідних даних. Для цього застосуємо швидке перетворення Фур'є [6]. Для множини чисел a_0, a_1, \dots, a_{n-1} швидке перетворення дозволяє отримати множину чисел b_0, b_1, \dots, b_{n-1} , таких, що

$$b_i = \sum_{j=0}^{n-1} a_j e^{ij},$$

де $e^n = 1$ і $e^k \neq 1$ при $0 < k < n$. У роботі застосовано реалізацію з Python-бібліотеки `numpy`, методи `numpy.fft.fft` та `numpy.fft.ifft` [7]. У даних методах швидке перетворення Фур'є для комплексних вхідних даних визначається як

$$A_k = \sum_{m=0}^{n-1} a_m e^{-2\pi i \frac{mk}{n}}, \quad k = 0 \dots n-1.$$

Після перетворення сигналу і його фільтрації визначається показник варіабельності серцевого ритму. Даний показник може бути обчислений різними методами, як статистичним, так і геометричними [4]. У даній роботі визначення варіабельності серцевого ритму реалізовано за через обчислення RMSSD – середньоквадратичного відхилення величини послідовних інтервалів R-R (інтервалів між зубцями R EKT). Воно визначається наступним чином. Нехай $IR_1, IR_2 \dots IR_{n-1}$ – інтервали між парами зубців $(R_2 - R_1), (R_3 - R_2), \dots, (R_n - R_{n-1})$ відповідно. Тоді

$$RMSSD = \sqrt{\sum_{k=1}^{n-2} \frac{(IR_{k+1} - IR_k)^2}{n-2}}.$$

У даній роботі такими інтервалами вважаємо інтервали між локальними максимумами фільтрованого сигналу. Обчислені значення RMSSD у мс дозволяють фахівцю інтерпретувати дані та робити висновок про стан серцево-судинної системи.

Крім варіабельності серцевого ритму за даними, отриманими через швидке перетворення Фур'є, також обчислюється частота серцевих скорочень (на хвилину). Частота серцевих скорочень N_{pm} з даних визначається за наступною формулою

$$N_{pm} = \frac{30}{t} (N_{max} + N_{min}),$$

де t – тривалість запису, с; N_{max}, N_{min} – кількість локальних екстремумів фільтрованого сигналу (максимальних та мінімальних піків відповідно).

Після визначення всіх показників результати виводяться в графічному вигляді. Для побудови графіків застосована графічна бібліотека `matplotlib` [8].

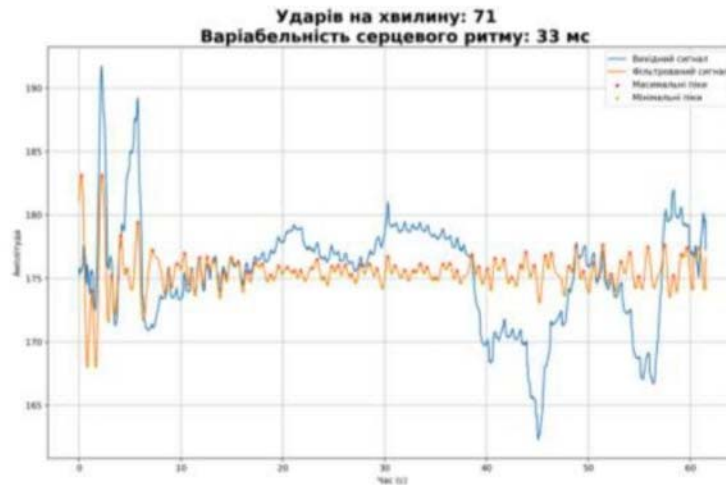


Рис. 2. Результат роботи програмного забезпечення

На рис. 2 видно, що програма виводить графіки вхідного оригінального сигналу з камери, та фільтрованого сигналу за допомогою перетворення Фур'є. Крім того, на графік виведено показники варіабельності серцевого ритму та частоти серцевих скорочень. Зауважимо, що дані графіки не можна інтерпретувати як електрокардіограму. Вихідний графік показує амплітуду зміни яскравості зображення в часі, а фільтрований сигнал – частотну складову.

Методика роботи з програмою наступна. Спочатку потрібно записати відеофайли, які будуть аналізуватись програмою. Це робиться за допомогою камери смартфона шляхом притискання пальця до його камери. Бажано максимально закрити пальцем камеру, щоб на запис не впливали перепади зовнішнього освітлення. Як відзначено при спостереженнях, при притисканні пальця до камери, у перші секунди спостерігається деяка осциляція сигналу, що може негативно вплинути на інтерпретацію результату. Потім сигнал стабілізується, тому рекомендується для підвищення точності починати запис через декілька секунд після початку притискання пальця для стабілізації кровообігу, а також проводити заміри тривалістю не менше 1 хв.

Для оцінки коректності вимірювань можна порівняти результати, отримані за допомогою програми з результатами, отриманими за допомогою смарт-пристроїв, які мають функцію визначення частоти серцевих скорочень (вимірювання пульсу). Було проведено 50 записів з камери смартфона з одночасним вимірюванням пульсу за допомогою Apple Watch 6 та Xiaomi Amazfit Verge. Тестова група включала осіб різного віку, статі та з різним рівнем здоров'я серцево-судинної системи. Заміри проводились протягом однієї хвилини, після години перебування в спокої, і в одному й тому ж приміщенні.

Результати вимірів показано на рис. 3. З метою покращення візуального сприйняття показано тільки перші 15 результатів. Як видно з графіку, показники отримані з камери смартфона та отримані за допомогою смарт годинників відрізняються несуттєво.

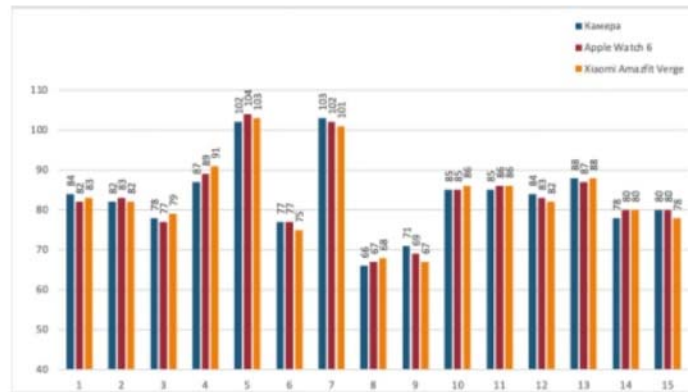


Рис. 3. Порівняння результатів вимірювань, отриманих із різних пристроїв

На рис. 4 наведено графіки середнього абсолютного відхилення значень частоти серцевих скорочень, отриманих за допомогою камери смартфона, від показників смарт годинників, а також відносна похибка.

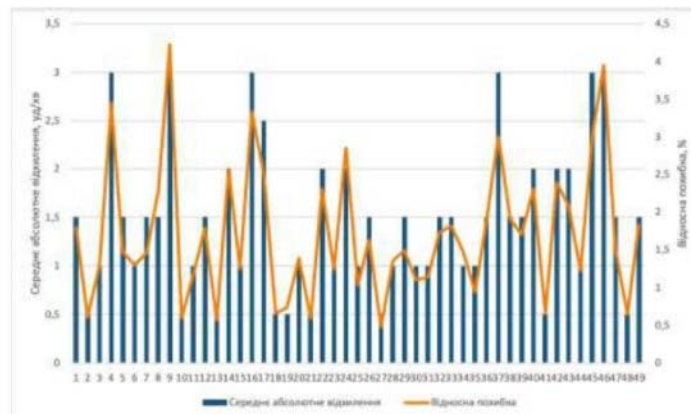


Рис. 4. Середнє абсолютне відхилення та відносна похибка отриманих результатів

Як бачимо з графіків, середнє абсолютне відхилення частоти серцебиття та відносна похибка знаходяться в межах 3-4% відсотків для тестової групи. В цілому можна сказати, що показники, які отримані за допомогою камери смартфона, є достатньо близькими до показників, які отримані за допомогою смарт годинників. Звертаємо увагу, що з медичної точки зору результати не можуть бути використані для постановки діагнозу, і для точного діагнозу необхідно звертатись до лікаря та проводити клінічне обстеження.

Висновки та перспективи подальшого дослідження. У даній роботі описано особливості реалізації програмного забезпечення для моніторингу показників серцевої діяльності з використанням камери смартфона. Для обробки вихідного сигналу використовується швидке перетворення Фур'є. Отримані результати можуть бути застосовані для повсякденного спостереження за станом здоров'я. В подальшому розроблене програмне забезпечення можна модифікувати. Зокрема доцільно реалізувати пакетну обробку відеофайлів, а також запис результатів в базу даних з метою їх подальшого аналізу та обліку.

Список бібліографічного опису

1. World Health Organization. The top 10 causes of death [Електронний ресурс]. 2020. Режим доступу до ресурсу: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>.

© Ройко О.Ю., Пугач Т.В.

2. Коваленко, В. М. Стрес і серцево-судинні захворювання: сучасний стан проблеми." Український кардіологічний журнал, №1, 2015. с. 4-7.
3. Nascimento, B., Brant, L., Moraes, D., Ribeiro, A. Глобальне здоров'я та серцево-судинні захворювання. Український кардіологічний журнал, №4, 2015. с. 123-133.
4. Сичов О.С., Жарінов О.Й. Варіабельність серцевого ритму: фізіологічні механізми, методи дослідження, клінічне і прогностичне значення [Електронний ресурс]. Режим доступу до ресурсу: <https://compendium.com.ua/uk/glava-4-variabelnist-sertseвого-ritmu-fiziologichni-mehanizmi-metodi-doslidzhennya-klinichne-i-prognostichne-znachennya/>.
5. OpenCV: Home [Електронний ресурс]. Режим доступу до ресурсу: <https://opencv.org/>
6. James W. Cooley, John W. Tukey: An algorithm for the machine calculation of complex Fourier series. In: Math. Comput. 19, 1965, p.p. 297—301.
7. [numpy.fft.fft](https://numpy.org/doc/stable/reference/generated/numpy.fft.html) [Електронний ресурс]. Режим доступу до ресурсу: <https://numpy.org/doc/stable/reference/generated/numpy.fft.html>
8. Matplotlib — Visualization with Python [Електронний ресурс]. Режим доступу до ресурсу: <https://matplotlib.org/>

References

1. World Health Organization. The top 10 causes of death [Electronic resource]. 2020. Mode of access to the resource: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>.
2. Kovalenko, V. M. Stress and cardiovascular diseases: the current state of the problem." Ukrainian Journal of Cardiology, No. 1, 2015. p. 4-7.
3. Nascimento, B., Brant, L., Moraes, D., Ribeiro, A. Global health and cardiovascular disease. Ukrainian Journal of Cardiology, No. 4, 2015. с. 123-133.
4. Sychev O.S., Zharinov O.Y. Heart rate variability: physiological mechanisms, research methods, clinical and prognostic significance [Electronic resource]. Mode of access to the resource: <https://compendium.com.ua/uk/glava-4-variabelnist-sertseвого-ritmu-fiziologichni-mehanizmi-metodi-doslidzhennya-klinichne-i-prognostichne-znachennya/>.
5. OpenCV: Home [Electronic resource]. Resource access mode: <https://opencv.org/>
6. James W. Cooley, John W. Tukey: An algorithm for the machine calculation of complex Fourier series. In: Math. Comput. 19, 1965, p.p. 297—301.
7. [numpy.fft.fft](https://numpy.org/doc/stable/reference/generated/numpy.fft.html) [Electronic resource]. Resource access mode: <https://numpy.org/doc/stable/reference/generated/numpy.fft.html>
8. Matplotlib — Visualization with Python [Electronic resource]. Resource access mode: <https://matplotlib.org/> Roger C. Palmer. The Bar Code Book. Trafford Publishing, 2007.

Додаток Б

Програмний код модуля main.py

```

from tkinter import *
from tkinter import filedialog as fd
import tkinter.ttk as ttk

import base as bs
import prePost as pp
import shutil
import os
import datetime

def new_test_person():
    bs.set_current_person_id_null()
    filetypes = (
        ('Відеофайли', '*.mov'),
        ('Відеофайли', '*.avi'),
        ('Відеофайли', '*.mp4'),
        ('Всі файли', '.*.*')
    )
    ts = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
    filename = fd.askopenfilename(
        title='Відкрийте відеофайл',
        initialdir='.',
        filetypes=filetypes)
    input_vid = './input_vid/' + os.path.basename(filename)
    shutil.copy(filename, input_vid)
    HR, HRV, output_img = pp.cardiotest(input_vid)
    add_new_person_ww()
    test_id =
bs.insert_person_test(bs.get_current_person_id(), ts, HRV,
HR, output_img)
    open_test_result_ww(test_id)

def add_new_person_ww():
    child_new_test = Toplevel(root)
    child_new_test.title("Додати особу")
    x = root.winfo_x()
    y = root.winfo_y()
    child_new_test.geometry("+%d+%d" %(x+200,y+0))

```

```

Label(child_new_test, text="Внесіть дані про нову
особу",font=('Arial',14)).grid(row=0, column=0, sticky=W+E,
columnspan=2)
Label(child_new_test, text="Прізвище").grid(row=1,
column=0, sticky=W)
ent_pr = Entry(child_new_test, width=30, fg='black',
font=('Arial',12))
ent_pr.grid(row=2, column=0, sticky=W+E, columnspan=2)
Label(child_new_test, text="Ім'я").grid(row=3, column=0,
sticky=W)
ent_im = Entry(child_new_test, width=30, fg='black',
font=('Arial',12))
ent_im.grid(row=4, column=0, sticky=W+E, columnspan=2)
Label(child_new_test, text="По батькові").grid(row=5,
column=0, sticky=W)
ent_pob = Entry(child_new_test, width=30, fg='black',
font=('Arial',12))
ent_pob.grid(row=6, column=0, sticky=W+E, columnspan=2)
Label(child_new_test, text="Дата
народження").grid(row=7, column=0, sticky=W)
ent_data_nar = Entry(child_new_test, width=30,
fg='black', font=('Arial',12))
ent_data_nar.grid(row=8, column=0, sticky=W+E,
columnspan=2)
Label(child_new_test, text="Стать").grid(row=9,
column=0, sticky=W)
ent_stat = Entry(child_new_test, width=30, fg='black',
font=('Arial',12))
ent_stat.grid(row=10, column=0, sticky=W+E,
columnspan=2)
Label(child_new_test, text="Чи мала\має особа
захворювання серцево-судинної системи").grid(row=11,
column=0, sticky=W)
ent_has_dis = Entry(child_new_test, width=30, fg='black',
font=('Arial',12))
ent_has_dis.grid(row=12, column=0, sticky=W+E,
columnspan=2)
Label(child_new_test, text="Додаткова
інформація").grid(row=13, column=0, sticky=W)
text_dod = Text(child_new_test, width=30, height=5,
fg='black', font=('Arial',12))
text_dod.grid(row=14, column=0, sticky=W+E,
columnspan=2)
Label(child_new_test, text="Додати особу в
базу").grid(row=15, column=0, sticky=W)

btn_start_test = Button(child_new_test, text="Додати",
command=lambda: add_new_person_cb(child_new_test,
ent_pr.get(), ent_im.get(),
ent_pob.get(),ent_data_nar.get(), ent_stat.get(),
ent_has_dis.get(), text_dod.get(1.0, "end-1c")))
btn_start_test.grid(row=15, column=1, sticky=W+E)
root.wait_window(child_new_test)

```

```

def add_new_person_cb(ww, prizv, im, pob, data_nar, stat,
have_dis, dod):
    cp = bs.insert_new_person(prizv, im, pob, data_nar, stat,
have_dis, dod)
    bs.set_current_person_id(cp)
    ww.destroy()

def open_test_result_ww(test_id):
    tdata, hrv, hr, tfile, pr, im, pob =
bs.select_test_person(test_id)
    child_test_result = Toplevel(root)
    x = root.winfo_x()
    y = root.winfo_y()
    child_test_result.geometry("+%d+%d" %(x+200,y+200))
    pib = ' '.join([pr, im, pob])
    child_test_result.title('Результат тесту: '+pib)
    Label(child_test_result,
text=pib,font=('Arial',16)).grid(row=0, column=0,
sticky=W+E, columnspan=2)
    Label(child_test_result, text="Дата та час тесту:
"+tdata).grid(row=1, column=1, sticky=E)
    Label(child_test_result, text="Частота серцевих
скорочень: "+str(hr)).grid(row=1, column=0, sticky=W)
    Label(child_test_result, text="Варіабельність серцевого
ритму: "+str(hrv)).grid(row=2, column=0, sticky=W)
    canvas = Canvas(child_test_result, width = 865, height =
450)
    canvas.grid(row=3, column=0, sticky=W+E,columnspan=2)
    img = PhotoImage(file=tfile)
    canvas.create_image(10,10, anchor=NW, image=img)
    root.wait_window(child_test_result)

def open_all_tests_persons_ww():
    child_tests_persons = Toplevel(root)
    x = root.winfo_x()
    y = root.winfo_y()
    child_tests_persons.geometry("+%d+%d" %(x+0,y+150))
    data = bs.select_all_tests_persons()
    columns = ("#1", "#2", "#3", "#4", "#5", "#6", "#7", "#8")
    tree = ttk.Treeview(child_tests_persons,
show="headings", columns=columns, height=30)
    tree.heading("#1",text="Прізвище")
    tree.column("#1",width=100)
    tree.heading("#2",text="Ім'я")
    tree.column("#2",width=100)
    tree.heading("#3",text="По батькові")
    tree.column("#3",width=100)
    tree.heading("#4",text="ID тесту")
    tree.column("#4",width=50)

```

```

tree.heading("#5",text="Дата тесту")
tree.column("#5",width=150)
tree.heading("#6",text="ВСП")
tree.column("#6",width=50)
tree.heading("#7",text="ЧСС")
tree.column("#7",width=50)
tree.heading("#8",text="Шлях до файлу")
tree.column("#8",width=200)
tree.grid(row=1, column=0)
for row in data:
    tree.insert("", END, values=row)
scroll = Scrollbar(child_tests_persons,
orient="vertical")
scroll.grid(row=1, column=1, sticky='ns')
scroll.config(command=tree.yview)
tree.config(yscrollcommand=scroll.set)

def show_selected_test(a):
    curItem = tree.focus()
    val = tree.item(curItem) ["values"]
    test_id = val[3]
    open_test_result_wv(test_id)

tree.bind('<<TreeviewSelect>>', show_selected_test)

root.wait_window(child_tests_persons)

def open_all_persons_wv():
    child_all_persons = Toplevel(root)
    child_all_persons.title("Persons")
    x = root.winfo_x()
    y = root.winfo_y()
    child_all_persons.geometry("+%d+%d" %(x+0,y+150))
    data = bs.select_persons()
    columns = ("#1", "#2", "#3", "#4", "#5", "#6", "#7")
    tree = ttk.Treeview(child_all_persons, show="headings",
columns=columns, height=30)
    tree.heading("#1",text="Прізвище")
    tree.column("#1",width=100)
    tree.heading("#2",text="Ім'я")
    tree.column("#2",width=100)
    tree.heading("#3",text="По батькові")
    tree.column("#3",width=100)
    tree.heading("#4",text="Дата народження")
    tree.column("#4",width=100)
    tree.heading("#5",text="Стать")
    tree.column("#5",width=50)
    tree.heading("#6",text="Чи має серцеві захворювання")
    tree.column("#6",width=50)
    tree.heading("#7",text="Додаткова інформація")
    tree.column("#7",width=150)
    tree.grid(row=1, column=0)
    for row in data:

```

```
        tree.insert("", END, values=row)
    scrollbar = Scrollbar(child_all_persons, orient="vertical")
    scrollbar.grid(row=1, column=1, sticky='ns')
    scrollbar.config(command=tree.yview)
    tree.config(yscrollcommand=scrollbar.set)
    root.wait_window(child_all_persons)
```

```
root = Tk()
root.title("Cardiotest")
btn_new_test = Button(root, text="Нове тестування",
command=new_test_person).pack(pady = 2)
btn_open_tests = Button(root, text="Перегляд бази тестів",
command=open_all_tests_persons_wd).pack(pady = 2)
btn_open_persons = Button(root, text="Перегляд бази осіб",
command=open_all_persons_wd).pack(pady = 2)
btn_quit = Button(root, text="Вийти",
command=root.destroy).pack(pady = 2)
root.attributes('-topmost', True)
root.mainloop()
```

Програмний код модуля prePost.py

```
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
import time
from sigMain import sigMain

def cardiostest(filename):
    cap = cv.VideoCapture(filename)

    total_frames = cap.get(cv.CAP_PROP_FRAME_COUNT)
    fps = cap.get(cv.CAP_PROP_FPS)
    duration = total_frames / fps
    ret = True

    # normal_sum_values = []
    hsv_v_values = []
    frame_x = []
    frame_c = 1

    converting_time = 0
    indexing_time = 0
    summedian_time = 0

    while ret:
        ret, frame = cap.read()

        if not ret:
            break

        frame_x.append(frame_c)
        frame_c += 1

        print(str(round(100*(frame_c/total_frames))) + " %")

        # GET HSV Value
        t = time.time()
        hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
        converting_time += time.time() - t

        t = time.time()
        hsv_values = hsv[:, :, 2]
        indexing_time += time.time() - t

        t = time.time()
        # print(hsv_values)
```

```
        mean_hsv_values      =      np.sum(np.sum(hsv_values,
axis=0),axis=0)/hsv_values.size
        summedian_time += time.time() - t

        hsv_v_values.append(mean_hsv_values)

# print("frame: " + str(frame_x))
# print("NORMAL SUM VALUES: ")
# print(len(normal_sum_values))

# print("HSV VALUES: ")
# print(len(hsv_v_values))

HR, HRV, output = sigMain(hsv_v_values, duration)
return HR, HRV, output
```

Програмний код модуля sigMain.py

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.signal import resample
from scipy.signal import find_peaks
from SNR import SNR

import time

def sigMain(sig, duration):

    sig = resample(sig, (len(sig) * 2) + int(0.1 * len(sig)))

    n = len(sig)

    dt = duration / n
    t = np.linspace(0, duration, n)

    fcomp = np.fft.fft(sig,n)
    PSD = fcomp * np.conj(fcomp) / n
    freq = (1/(dt*n)) * np.arange(n)
    #print(freq[0:45])
    L = np.arange(1, np.floor(n/2), dtype="int")

    # signalsTest.plotSpectrum(freq, PSD, L)
    # print(PSD[-1], PSD[1])
    # print(fcomp[1], fcomp[-1])
    # plt.plot(freq, fcomp)
    # plt.xlim(freq[0], freq[int(np.floor(len(freq)/2))])
    # plt.grid()

    ffilt = fcomp
    freqMin = 0.38 #Hz
    freqMax = 2 #Hz
    for f in range(1,int(np.floor(len(freq)/2))):
        tempFreq = freq[f]
        if tempFreq <= freqMin or tempFreq >= freqMax:
            ffilt[f] = 0
            ffilt[-f] = 0

    # print(ffilt)
    # print("original: ", fcomp)
```



```

# print("filtered: ",ffilt)
PSDfilt = ffilt * np.conj(ffilt) / n

sigfilt = np.real(np.fft.ifft(ffilt))
for nc in range(len(sigfilt)):
    sigfilt[nc] = sigfilt[nc].real

nInds = int(len(sigfilt)*0.02)      # percetage of padding
sigfilt[0:nInds] = sigfilt[nInds*2:nInds:-1]
sigfilt[-1-nInds:-1] = sigfilt[-1-(nInds*2):-1-nInds]

plt.rcParams['figure.figsize'] = [10, 4]

fig,axs = plt.subplots(1,1)

plt.sca(axs)
# --- PLOT ORIGINAL SIGNAL
plt.plot(t, np.real(sig), label='Вихідний сигнал')

# --- PLOT FILTERED SIGNAL
plt.plot(t, sigfilt, label="Фільтрований сигнал")

horzdist = 12
peaks = find_peaks(sigfilt, height=1, distance=horzdist)
height = peaks[1]['peak_heights']
peak_pos = t[peaks[0]]
plt.scatter(peak_pos, height, color="r",
label="Максимальні піки", marker=".")

# MIN PEAKS
invsig = sigfilt * -1
minima = find_peaks(invsig.real, distance=horzdist)
min_pos = t[minima[0]]
min_height = invsig[minima[0]]
min_height = min_height * -1
plt.scatter(min_pos, min_height, color="y",
label="Мінімальні піки", marker=".")

timePeaksms = t[peaks[0]]*1000

rrIntervals = [0 for i in range(len(timePeaksms)-1)]
for b in range(len(timePeaksms)-1):
    rrIntervals[b] = timePeaksms[b+1] - timePeaksms[b]

variabilities = [0 for i in range(len(rrIntervals)-1)]
for v in range(len(rrIntervals)-1):
    variabilities[v] = rrIntervals[v+1] - rrIntervals[v]

```

```

soma = 0
MSSD = [0 for i in range(len(variabilities))]
for s in range(len(variabilities)):
    soma = soma + (variabilities[s]**2)
    # print(soma)

rMSSD = np.sqrt(soma/len(variabilities))    #ms
# HRV = rMSSD
HRV = int(rMSSD/10)

# --- BPM ---
nbpMAX = len(peak_pos)
nbpMIN = len(min_pos)
nbp = (nbpMIN + nbpMAX) / 2
nbpm = int(np.around(60 * nbp / duration))

# --- SNR ---
SignalToNoiseRatio = abs((SNR(sig, sigfilt)).real) # dB

plt.legend()
plt.title("Ударів на хвилину: %i \n Варіабельність
серцевого ритму: %i мс " % (nbpm, HRV), fontdict={'fontsize':
20, 'fontweight': 'bold'})
plt.ylabel("Амплітуда")
plt.xlabel("Час (с)")
plt.grid()
ts = time.time()
outfile = 'output_img/' + str(int(ts*1000)) + '.png'
plt.savefig(outfile, bbox_inches='tight',dpi=100 )

return nbpm, HRV, outfile

```

Програмний код модуля base.py

```
import sqlite3

from sqlite3 import Error

database = r"db.db"

def create_connection():
    conn = None
    try:
        conn = sqlite3.connect(database)
    except Error as e:
        print(e)
    return conn

def set_option(key, value):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
UPDATE options SET value = ? WHERE key = ?
''', (value, key))
    conn.commit()
    conn.close()

def get_option(key):
    conn = create_connection()
    sql = 'SELECT value FROM options WHERE key = ?'
    cur = conn.cursor()
    cur.execute(sql, (key,))
    rows = cur.fetchall()
    conn.close()
    return rows[0][0]

def get_current_person_id():
    return get_option('current_person_id')

def set_current_person_id(id):
    set_option('current_person_id', id)

def set_current_person_id_null():
    set_option('current_person_id', 0)

def select_persons():
    conn = create_connection()
```

```

    sql = 'SELECT prizv, im, pob, data_nar, stat, have_dis,
dod FROM persons'
    cur = conn.cursor()
    cur.execute(sql)
    rows = cur.fetchall()
    conn.close()
    return rows

def select_person_test(person_id):
    conn = create_connection()
    sql = 'SELECT p.prizv, p.im, p.pob, t.created_at, t.hrv,
t.hr, t.img_path FROM persons AS p JOIN cardioteests AS t ON
p.id = t.person_id WHERE p.id = ?'
    cur = conn.cursor()
    cur.execute(sql, (person_id,))
    rows = cur.fetchall()
    conn.close()
    return rows

def select_all_tests_persons():
    conn = create_connection()
    sql = 'SELECT p.prizv, p.im, p.pob, t.id, t.created_at,
t.hrv, t.hr, t.img_path FROM persons AS p JOIN cardioteests AS
t ON p.id = t.person_id'
    cur = conn.cursor()
    cur.execute(sql)
    rows = cur.fetchall()
    conn.close()
    return rows

def select_test_person(test_id):
    conn = create_connection()
    sql = 'SELECT t.created_at, t.hrv, t.hr, t.img_path,
p.prizv, p.im, p.pob FROM cardioteests AS t JOIN persons AS p
ON t.person_id = p.id WHERE t.id = ?'
    cur = conn.cursor()
    cur.execute(sql, (test_id,))
    rows = cur.fetchall()
    conn.close()
    return rows[0]

def insert_person_test(person_id, created_at, hrv, hr,
img_path):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
INSERT INTO cardioteests (person_id, created_at, hrv, hr,
img_path)
VALUES (?, ?, ?, ?, ?)
''', (person_id, created_at, hrv, hr, img_path))
    conn.commit()

```

```

last_row = cur.lastrowid
conn.close()
return last_row

def insert_new_person(prizv, im, pob, data_nar, stat,
have_dis, dod):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
INSERT INTO persons (prizv, im, pob, data_nar, stat,
have_dis, dod)
VALUES (?, ?, ?, ?, ?, ?, ?)
''', (prizv, im, pob, data_nar, stat, have_dis, dod))
    conn.commit()
    last_row = cur.lastrowid
    conn.close()
    return last_row

def insert_new_person_test(prizv, im, pob, data_nar, stat,
have_dis, dod, created_at, hrv, hr, img_path):
    last_row = insert_new_person(prizv, im, pob, data_nar,
stat, have_dis, dod)
    insert_person_test(last_row, created_at, hrv, hr,
img_path)

def update_person(id, prizv, im, pob, data_nar, stat,
have_dis, dod):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
UPDATE persons
SET prizv = ?, im = ?, pob = ?, data_nar = ?, stat = ?,
have_dis = ?, dod = ?
WHERE id = ?
''', (prizv, im, pob, data_nar, stat, have_dis, dod, id))
    conn.commit()
    conn.close()

def delete_person_all(person_id):
    conn = create_connection()
    cur = conn.cursor()
    cur.execute('''
DELETE FROM cardiotests WHERE person_id = ?
''', (person_id, ))
    conn.commit()
    cur.execute('''
DELETE FROM persons WHERE id = ?
''', (person_id, ))
    conn.commit()
    conn.close()

```